

# **Deep Learning Classification for Parkinson's Disease**

## **Using Segmentation Techniques**

**By**

**Dhruvit Nagar - 18BCE128**

**Chandni Tomar - 18BCE246**



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**Ahmedabad 382481**

# **Deep Learning Classification for Parkinson's Disease Using Segmentation Techniques**

## **Minor Project Report**

Submitted in partial fulfilment of the requirements

For the degree of

**Bachelor of Technology in Computer Science & Engineering**

By

**Dhruvit Nagar - 18BCE128**

**Chandni Tomar - 18BCE246**

Guided By

**Prof. Shivani Desai**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**Ahmedabad 382481**

## CERTIFICATE

This is to certify that the minor project entitled “Deep Learning Classification for Parkinson’s Disease Using Segmentation Techniques” submitted by Dhruvit Nagar - 18BCE128, Chandni Tomar - 18BCE246, towards the partial fulfillment of the requirements for the degree of Bachelor of Technology in Computer Science and Engineering of Nirma University is the record of work carried out by him/her under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination.



Prof. Shivani Desai,  
Assistant Professor,  
Computer Science and Engineering Dept.,  
Institute of Technology,  
Nirma University,  
Ahmedabad



Dr. Madhuri Bhavsar,  
Professor and HOD,  
Computer Science and Engineering Dept.,  
Institute of Technology,  
Nirma University,  
Ahmedabad

## **ACKNOWLEDGEMENT**

*We would like to express our special thanks of gratitude to our guide "Prof. Shivani Desai" for the able guidance and support throughout in completing our project.*

*We would also like to extend our gratitude to "Prof. Zunnun Narmawala", for providing us with all the necessary suggestions to incorporate in our work to make it better and impactful.*

## **ABSTRACT/ Outline**

Parkinson's disease (PD) is a neurological illness that causes a motor function to deteriorate owing to the loss of dopamine-producing brain cells, or neurons. Tremors, stiffness, sluggish movements, shaking, and loss of balance are among the most common symptoms of Parkinson's disease. In this research, we employed a variety of models to diagnose Parkinson's disease early. In recent years, Deep Learning (DL) algorithms are very useful for the diagnosis of PD as DL algorithms work well with large datasets. We have mentioned various methods and segmentation techniques that play an important role in increasing accuracy.

# **CONTENTS**

<b>Certificate</b>	I
<b>Acknowledgement</b>	II
<b>Abstract</b>	III
<b>Chapter 1</b> <b>Introduction</b>	1
<b>1.1. Motivation</b>	1
<b>1.2. Objectives</b>	2
<b>Chapter 2</b> <b>Literature Survey</b>	2
<b>2.1. Comparative study of different approaches</b>	5
<b>Chapter 3</b> <b>Methodology</b>	6
<b>3.1. Dataset description</b>	7
<b>3.2. Preprocessing</b>	8
<b>3.3. Why use segmentation?</b>	11
<b>3.4. Implementation</b>	15
<b>Chapter 4</b> <b>Result Analysis</b>	21
<b>Chapter F</b> <b>Conclusion and Future work</b>	23
<b>Conclusion</b>	
<b>Future work</b>	
<b>References</b>	

## **1. Introduction**

Parkinson's disease (PD) is a chronic neurological illness that affects the human brain's function and is incurable. Patients are given various medicines to help halt the growth of the disease and to provide temporary relief from the symptoms. The substantia nigra, a component in the basal ganglia that is located in the midbrain, includes a substantial number of dopamine neurons.

This condition is caused by the thalamic region's neurons deteriorating, resulting in tremor resting, bradykinesia, and patient stiffness difficulties. Exhaustion, nausea, insomnia, difficulty speaking, a slowed rate of thoughts, and a speech disorder are some of the symptoms. After Alzheimer's disease, Parkinson's disease is the second most frequent neurological condition among the elderly.

The primary origin of Parkinson's disease is unknown at this time, while genetic and environmental factors are suspected. Due to a lack of diagnostic facilities, it is usually treated at a late stage. Doctors make decisions based on history and medical science. Because there are common indications in different neurodegenerative disorders, this technique is unreliable. It is usually diagnosed when dopamine hormones are irreparably damaged. It's still tough to make a precise PD diagnosis. If a patient with Parkinson's disease is treated as if he or she is stable and healthy, the disease might progress and become difficult to manage. Although some health studies can detect it, because it is linked to biochemical changes in the brain, identification through visual analysis is a more effective method.

### **1.1. Motivation**

Parkinson's disease (PD) is a neurodegenerative condition characterised by the premature death of dopaminergic neurons in the substantia nigra [1]. Our goal is to create a convolutional neural network-based deep learning model that can extract important features from MRI images and correctly categorise healthily and Parkinson's

patients. Due to various applications of segmentation techniques in the field of image processing in DL.

## 1.2. Objective

Our primary objective in this project is to develop a deep learning classification model using a variety of methods, including segmentation and non-segmentation methods, that can significantly improve the model's accuracy and identify essential features from MRI images, as well as classify MRI images in healthy and Parkinson's disease patients with greater accuracy.

Before a deep learning model can categorise an MRI image as healthy or Parkinson's disease patients, the images must first be preprocessed to remove unwanted noise and to balance and remove some intensity levels. The dataset included NIFTI format image files from the PPMI database, which included two types of images from an MRI scan of the human brain: flair and non-flair (T2 weighted). The images are 3-Dimensional in nature. The images are in grayscale and only represent the axial view. After that, segmentation is used to collect some of the significant features in front of the images, which can aid the deep learning model in learning more about the important features and making classification decisions. For a better understanding of the outcome, various result and assessment metrics such as accuracy, loss, and confusion matrix are displayed.

## 2. Literature review

### *Early detection of Parkinson's disease using image processing and artificial neural network. [3]*

The information photographs in this research are also from the PPMI vault. A total of 200 mind-checked images are used (including both healthy and Parkinson's-related images). The information images are SPECT (Single Photon Emission Computed Tomography) images. They used a variety of pre-preparing techniques. On the dataset, spatial standardisation, binary veil, and unsharp covering strategies are used to improve the effectiveness, resulting in greater execution and precision. A

comprehensive learning network was used to prepare the model. Most importantly, the information images are divided. The model is then created using this design, which is based on a counterfeit neural organisation with the sigmoid capacity as the initial work. So far, they've achieved a 94 percent accuracy rate. The problem of this study is that low-resolution SPECT images are used, which may result in poor model execution and compromise the finding method.

### ***MRI Segmentation of the Human Brain: Challenges, Methods, and Applications. [4]***

This paper explains why segmentation is significant in magnetic resonance imaging of the human brain in a brilliant way. Image division is often used in mind MRI research for estimating and visualizing the cerebrum's physical constructions, checking mind modifications, portraying obsessive regions, and carefully organising and picture directed mediations. Various division methods of varying exactness and degree of complexity have been developed and discussed in the writing during the last few years. They have outlined the distinctions between several MRI segmentation approaches, as well as their capabilities, benefits, and drawbacks. For the MRI preprocessing processes, they used image registration, bias field correction, and the removal of non-brain tissue. Manual segmentation, intensity-based segmentation, atlas-based segmentation, surface-based approaches, and hybrid segmentation methods are all discussed in this study, along with their benefits and drawbacks.

### ***Detection of Parkinson Disease in Brain MRI using Convolutional Neural Network.[5]***

It depicts a mechanised indication structure based on the Convolution Neural Network (CNN), which distinguishes between Parkinson's and healthy patients. The data was extracted from the PPMI collection and used in the model's development. Complete 500 T2 weighted MRI tests using a DICOM (Digital Imaging and Communications in Medicine) design, with 250 Parkinson's outputs for the MRI and 250 for the Healthy personnel. In accordance, information is divided into three sections: preparation, approval, and testing, which constitute 70%, 10%, and 20% of the obtained images,

respectively. X-rays are converted from DICOM to JPEG format using the DICOM-to-JPEG programming bundle. The 100x100 midbrain window was trimmed to access the key district of substantia nigra and was finally embedded into CNN, which helps reduce computational expense.

The proposed system captures input MR images, which are then labelled as PD or HC. The model remembers eight fundamental layers: two convolution layers, one after the other, each followed by a max pool layer, and a third convolution layer, followed by two entirely associated layers and a yield layer. The part size of both convolution and max-pool layers is 3x3, resulting in 32 component maps. The CNN learns these component maps, allowing it to distinguish between PD and HC MRI configurations. Various tests for various organisational designs are run, including changing or adding different hyperparameter variables such as cushioning (either same or legitimate), piece size, step length, and cluster size. The dropout rate is included to reduce overfitting while keeping track of irrelevant information. In the two test results, order exactness ranges from 95 to 98 percent. On the same dataset, the suggested model outperforms SVM, Decision Tree, RVM, ANN, and other machine learning algorithms in terms of precision.

### ***Deep learning based diagnosis of Parkinson's disease using convolutional neural network. [2]***

They employed a deep learning approach called neural networks to assess the difference between a stable or healthy person's magnetic resonance images and those of a person with Parkinson's disease. To improve parkinson's disease detection, they deployed the AlexNet deep learning neural network, which takes advantage of convolutional neural network architecture.

They employed MR images and deep learning techniques to train the model, and transfer learning was applied to improve understanding and improve the model's accuracy. The researchers have employed many types of PPMI brain MRI images. The images are T2-weighted images in the axial plane derived from the PPMI's public domain database, which includes both parkinson's patients and healthy subjects and is directly sampled from one database to maintain uniformity.

They used 80% of the images as the preparation dataset and 20% of the images as the testing dataset. The preparing dataset has 2826 HC images and 3296 PD images, while the testing dataset contains 705 HC images and 825 PD images. A 2D Gaussian Filter is used to remove clamour from the images before they are processed. As information, a pre-trained AlexNet with 227 X 227-pixel shading photos is provided, and computations are performed in each layer until a class yield is obtained. Using stochastic slope drop calculation, the organisation loads in the totally associated layer are tuned during handling. The recommended AlexNet engineering achieved an accuracy of 88.90 percent. The true positive rate and genuine negative rate upsides are 89.30 percent and 88.40 percent, respectively, in this design.

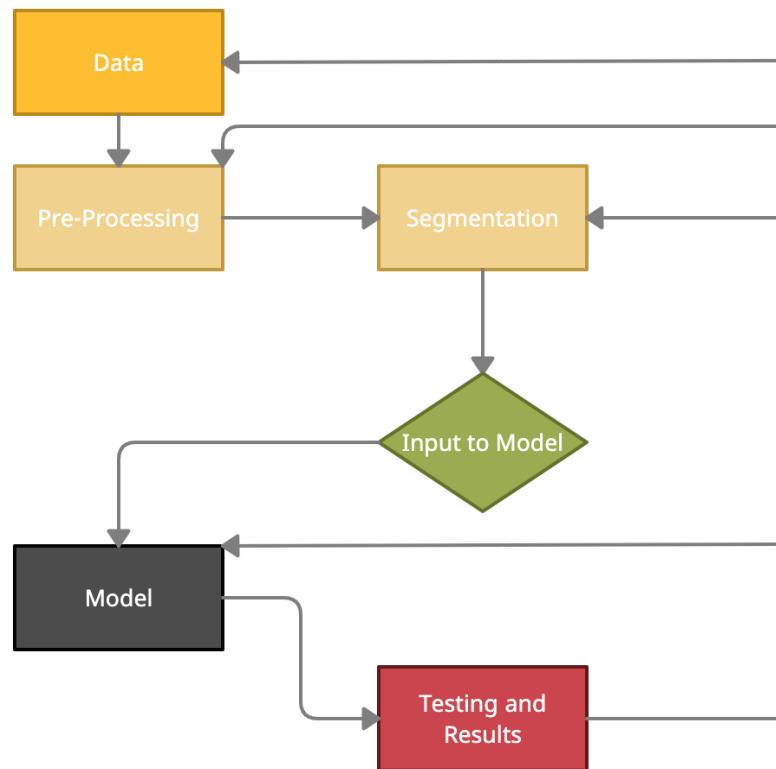
## 2.1. Comparative study of different approaches

<b>Ref. No</b>	<b>Author (year)</b>	<b>Technique Applied</b>	<b>Highest Accuracy</b>
6	K.Akyol (2020)	7-layer DNN model having a number of nodes as 2, 4, 8, 16, 32, 64 and 64 respectively with <i>SGD optimization</i> and <i>Relu activation function</i>	95.15%
7	S. Kaur et al (2020)	<i>To construct a deep learning prediction model Parkinson's illness, researchers used grid search optimization.</i>	91.69%
8	D. R . Rizvi et al (2020)	<i>DNN</i> and <i>LSTM</i> models. Different experiments were done on a dataset to find the optimal parameters of the network.	99.03%
9	Tuncer et al (2020)	The minimum average maximum tree was used along with a combination of singular valued decomposition. On the original dataset, the top 50 features were selected by the relief feature selection method. The <i>k-NN</i> was used as a classifier with a 10-fold CV. Post-processing was also done to obtain the individual results.	96.83%
10	Savitha S. Upadhyaya et al (2018)	The spectrogram characteristics and phonation of speech were extracted. As a neural network (NN) classifier, they employed a feed-forward network with two	98.0%

		layers and one hidden layer. A scaled conjugate backpropagation technique was utilised throughout the network's training. A softmax function was employed to do classification in the output layer.	
2	S. Sivaranjini (2019)	Convolution neural network AlexNet is used. They have trained models by the transfer learning network. In preprocessing they have normalized the images and 2D Gaussian filter for noise reduction is used. standard deviation $\sigma$ of 0.8.	88.9 %
5	Shah et al.. (2018)	Customized CAD based CNN architecture to classify MRI patches of Parkinson and healthy patterns, with 3 convolution layers learns the training sample of PPMI dataset	95-98 %
11	B. Peng et al. (2016)	<i>SVM</i> is used as a machine learning algorithm and Multilevel ROI based features are also used to improve the accuracy. They have used MRI images	92.35%
12	G. Pahuja and T. N. Nagabhushan (2016)	They have provided a novel approach to classify whether the person has Parkinson's disease or not using MRI images. <i>ELM-based</i> method along with <i>Genetic Algorithm</i> (GA-ELM) feature subset selection has been proposed.	89.22%
3	Rummanet et al.. (2018)	They have used the <i>Artificial neural network</i> and image processing on 100 PD and 100 NC images	94 %
13	Shivangi, Anubhav Johri and Ashish Tripathi (2019)	Parkinson Disease Detection Using <i>Deep Neural Networks</i>	VGFR Spectrogram Detector (88.1%) Voice Impairment Classifier (89.15%)

### 3. Methodology

We have followed the below approach in the entire project. A detailed description of each is provided in this paper.

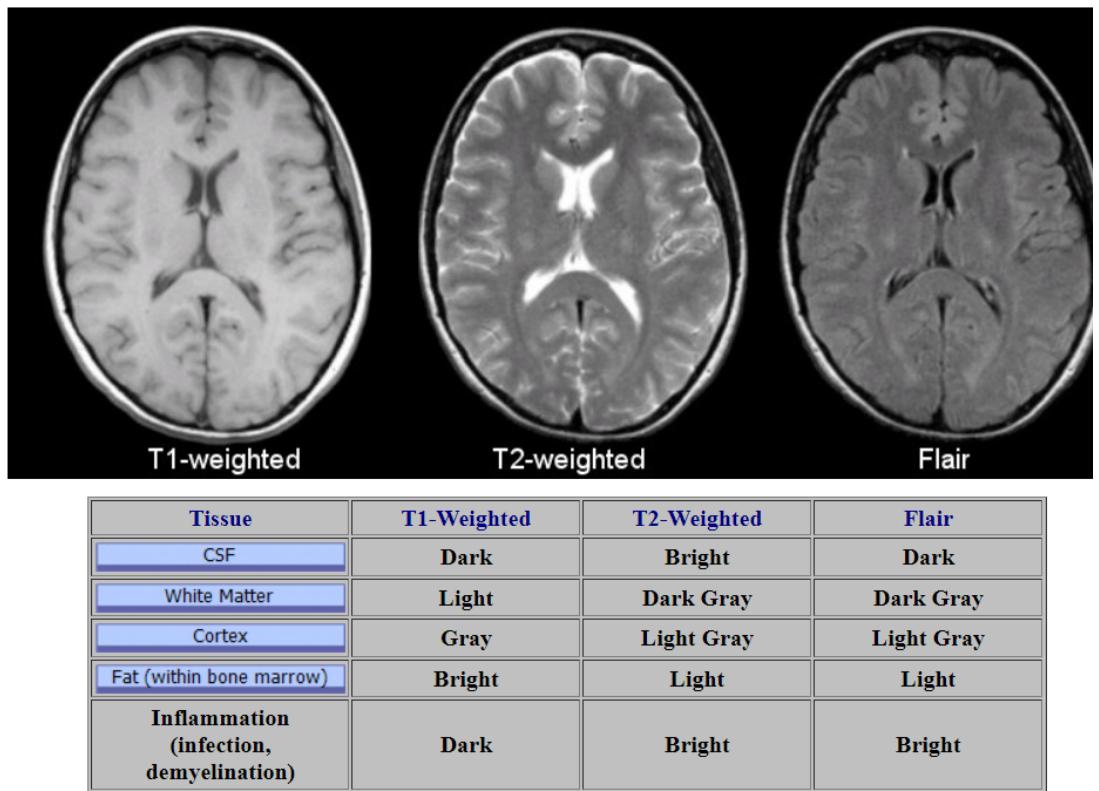


*Fig 1. Approach diagram*

We have followed the below approach in the entire project. A detailed description of each is provided in this paper.

### **3.1. Dataset Description**

Data is collected from the PPMI database in the form of Nifti (.nii) images, these include Axial viewed Flair and T2 weighted MRI scans. The Parkinson's Progression Markers Initiative (PPMI) classifies bio-markers of Parkinson's disease progression utilising digital imaging, biological screening, and clinical and quantitative trials. PPMI is hosted at several hospital locations in the United States, Germany, Israel, and Australia. Evidence and observations from test individuals have helped to build a strong Parkinson's database and biorepository, which is now available to the medical community for field-change analysis. The data (images) will be stored in Nifti (.nii) format and will include 77 healthy and 223 PD brain MRI scans for training, validation, and testing.



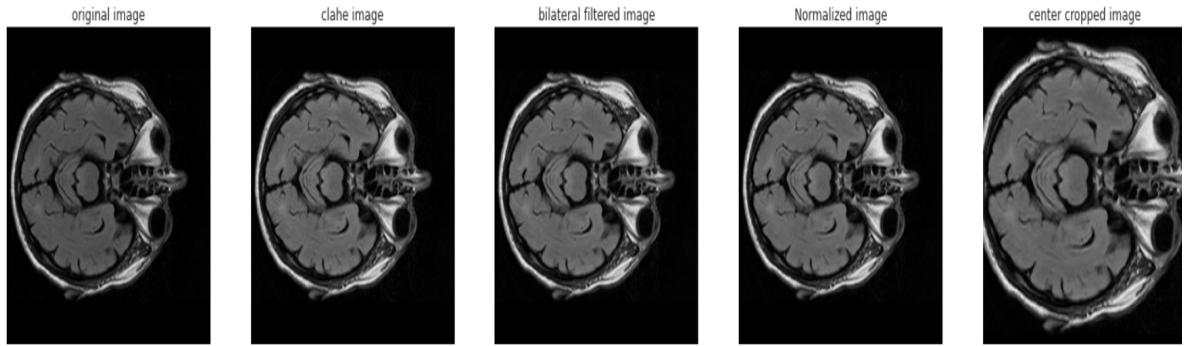
*Fig 2 . Types of brain MRI-scanned images*

### 3.2. Preprocessing

It is necessary to ensure that an image dataset is uniform and compatible before using it as an input to a deep learning model. In the case of disease prediction, data pre-processing also assists in improving the model's accuracy, resulting in improved performance and diagnosis. Resizing the images, removing bias from the images to make them uniform, normalisation of the images, segmentation of the images, standardising the frequency of the pixels in the images, cropping, data augmentation, and so on are all examples of pre-processing the dataset. Some particular image processing approaches are applied in the case of neuroimages. The basic pre-processing tasks used in this project are as follows:

- N4 bias field correction
- Image resizing

- c. Contrast limited adaptive histogram equalization
- d. Bilateral filter
- e. Image normalization
- f. Centre cropping the image



**Fig 3.** Sample images after applying various pre-processing

The entire dataset is then split into two parts: a training set with 80% of the data and a validation set with 20% of the data. Similarly, the class labels are separated into two categories: training and validation. After applying data augmentation, the model was trained on the training set and then confirmed on the validation set after each epoch.

The preprocessing pipeline for neuroimages includes phases such as field inhomogeneity correction, motion correction, registration, and segmentation, to name a few.

### 1) Magnetic Field Inhomogeneity Correction

The three components that make up the brain are grey matter (GM), Cerebrospinal fluid (CSF) & white matter (WM) (CSF). Although the magnetic field within the scanner should be consistent, when it comes into touch with brain tissue, it becomes variable. It decreases, and the rate of decrease varies depending on the tissue type. This occurrence is known as magnetic field inhomogeneity. These differences in magnetic field magnitude produce unnaturally bright and dark regions, It's tough to tell where tissue ends and another begins, making it difficult to discern tissue borders.

As a result, before removing non-brain tissue and segmenting the tissue, a magnetic field distributed system should be addressed.

## **2) Motion Correction and Volume Scrubbing**

Participants are not allowed to move their heads during an MRI scan.

Head motions, on the other hand, are unavoidable, and the data suffers as a result of motion-related distortions. As a result, the head motion should be corrected in all MRI data.

A rigid-body transformation is being used to register all dimensions to a standard volume and rectify motion. Any volume can be used as a reference volume, however, it is commonly the beginning or middle volume of the complete data set. The second stage is to minimise volume by moving your head severely. This technique is known as volume scrubbing.

## **3) Non-brain Tissue Removal**

The brain is the region of interest (ROI) for neuroimaging research. Non-brain tissues like the skull, throat, eyes, nose, and mouth are therefore irrelevant. When removing non-brain tissue, the gradient of intensity levels in different types of tissue is taken into account.

## **4) Intensity Normalization**

Different MRI data may have different ranges of intensity levels because MRI data does not have a defined unit. To normalise the range of intensity values throughout all 4D volumes with a single value, intensity normalisation is used.

## **5) Temporal Filtering**

The low-frequency range is known to be the region of interest in MRI data. Extremely low-frequency signals, on the other hand, are considered slow drifts (non-neuronal signals). As a result, bandpass filtering is commonly utilised to capture the desired

signals. Cutoff frequencies vary significantly between research, although filter ranges of 0.008-0.09 Hz and 0.01-0.1 Hz are commonly used.

## **6) Spatial Smoothing**

Spatial smoothing has the advantage of minimizing noise, but it also has the disadvantage of lowering signal intensity. As a result, while using spatial smoothing, researchers must proceed cautiously.

## **7) Registration**

The process of matching photographs of distinct geometric spaces in a single area is known as registration. The registration process has three basic components. The first step is to define a spatial-geometric transformation. Translation, rotation, scale, and shear in the x, y, and z dimensions are the 3D transformation parameters. Rigid body transformation. Second, a cost function must be specified that measures the goodness of the alignment. Finally, an interpolation method must be specified.

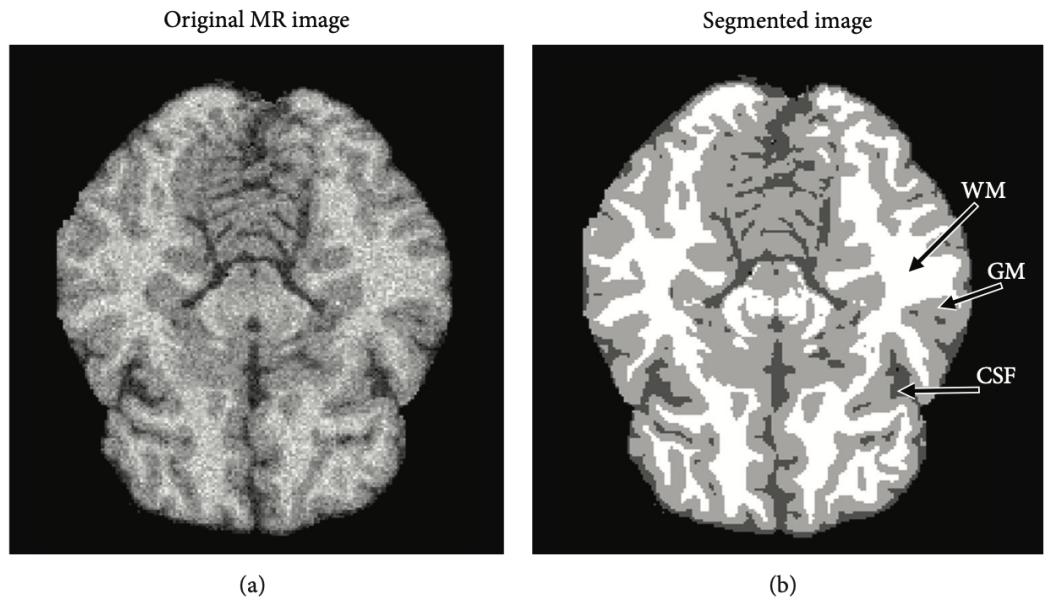
## **8) Segmentation**

The oscillations in time series in GM have been linked to neuronal signals, whereas those in WM and CSF have been linked to artefacts. In order to extract signals of interest, it is necessary to discriminate between GM, WM, and CSF tissues. This is done using segmentation.

### **3.3. Why use Segmentation?**

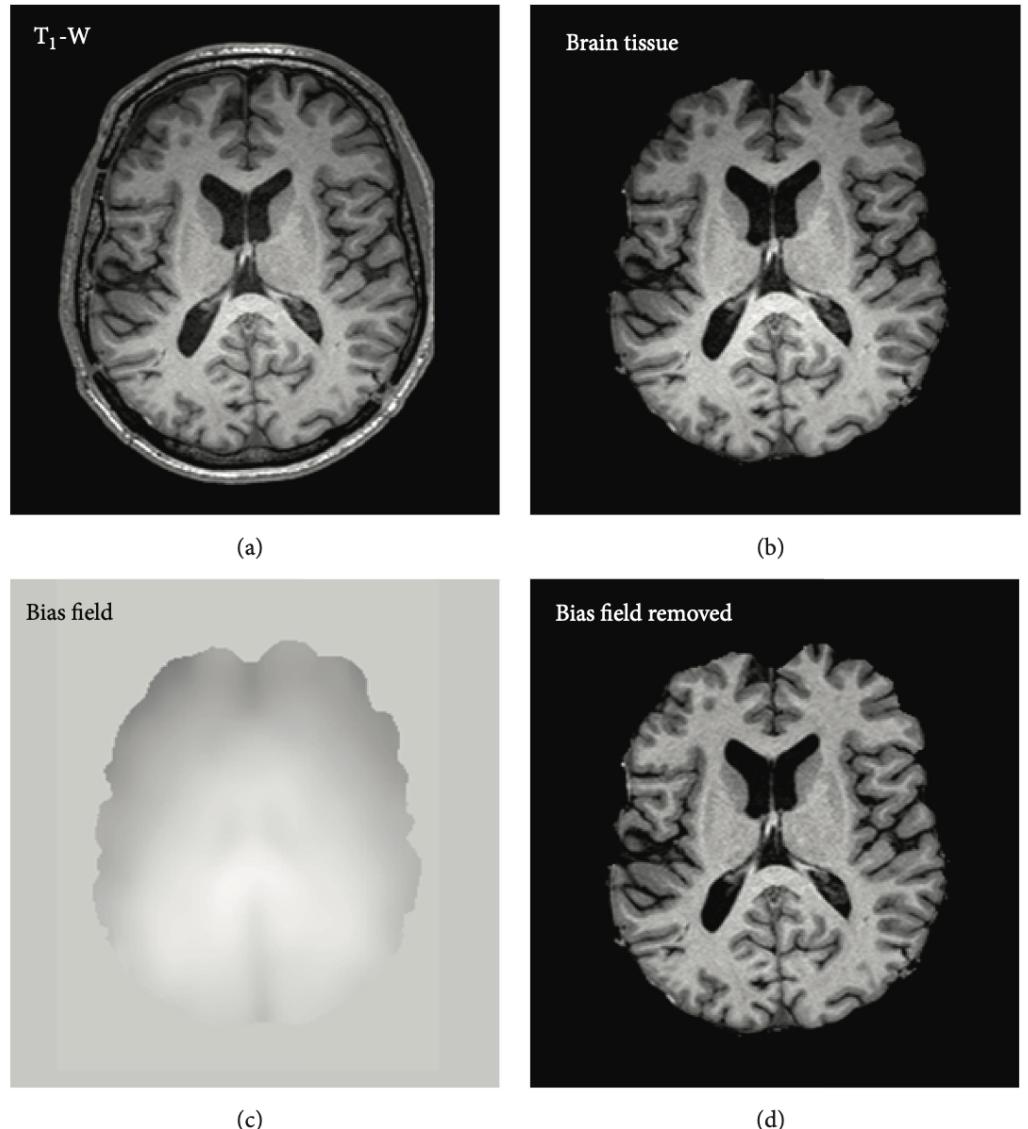
Image division is one of the main undertakings in clinical picture examination and is regularly the first and the most basic advance in numerous clinical applications. In cerebrum MRI investigation, image division is normally utilized for estimating and envisioning the mind's physical designs, for dissecting cerebrum changes, for portraying neurotic locales, and for careful arranging and picture directed mediations. Over the most recent couple of years, different division procedures of various precision and level of intricacy have been created and detailed in the writing.

The objective of image segmentation is to partition a picture into a bunch of semantically meaningful, homogeneous, and non-overlapping areas of comparable traits like intensity, profundity, shading, or surface. The division result is either a picture of marks recognizing each homogeneous area or a bunch of Forms that illustrate the boundaries of a location. The segmentation data are then used for a variety of purposes, including anatomical structure analysis, pathological area investigation, surgical planning, and visualisation[4].



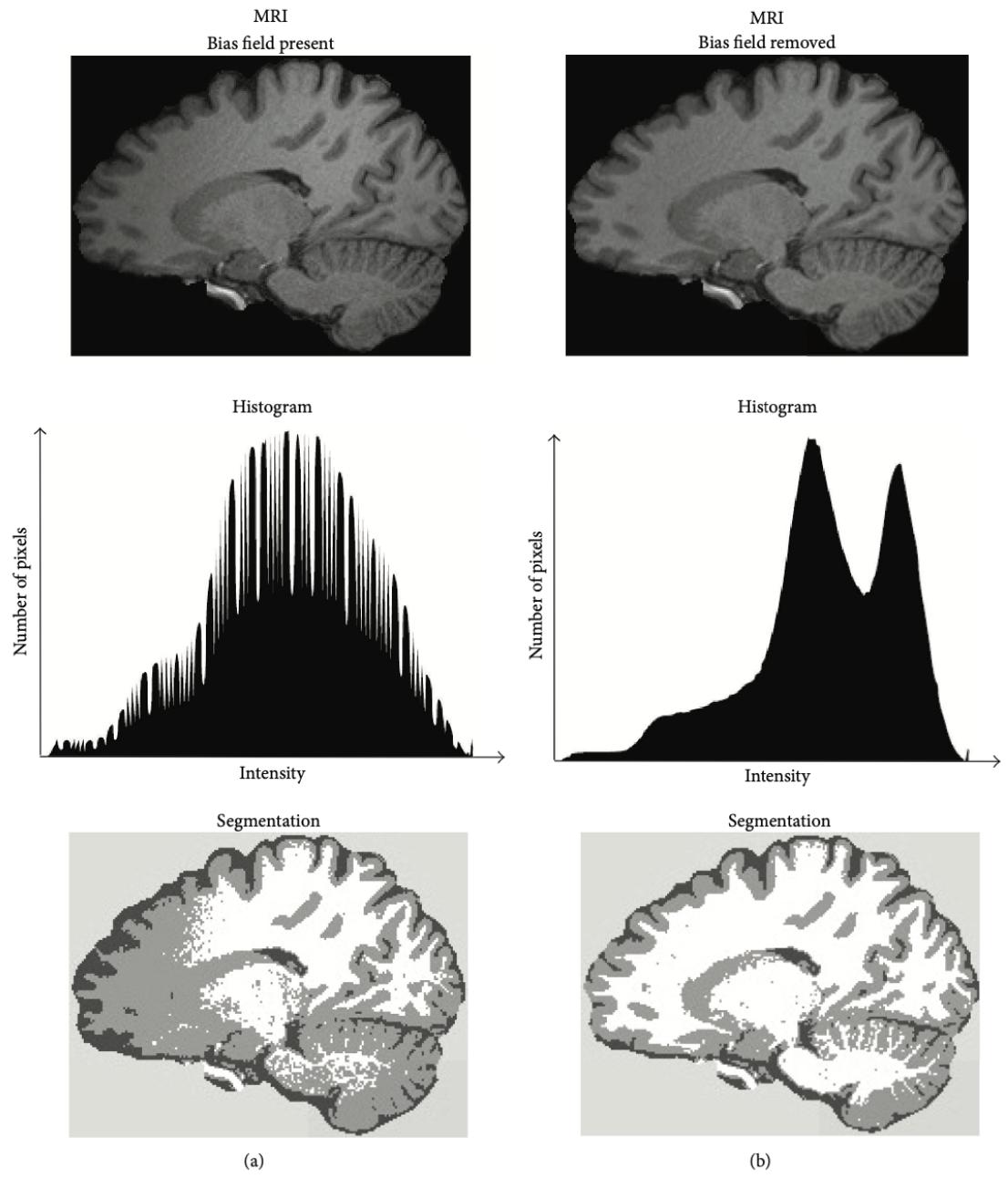
**Fig 4.** With an initial MR picture(a) and a segmented image with three labels, an instance of brain MRI segmentation: WM, GM, and CSF (b).

Image registration, bias field correction, and non-brain tissue removal are all processes in the MRI preprocessing process. After the pictures have been preprocessed, image segmentation is used to improve the accuracy of the deep learning model.



**Fig 5.** (a) the adult brain's original T1-W MR image; (b) the brain tissue image after deleting non-brain features; (c) the bias field; (d) the brain tissue image after bias field correction

The importance of preprocessing the MRI images can be seen below, it's important to preprocess the image because in some image segmentation methods the method itself produces the different output for the images containing noise and images containing without noise. So it's better to know it beforehand.



**Fig 6.** The bias field's effect on brain MRI segmentation. (a) The top of the graphic shows an example of a sagittal brain MRI slice with a bias field. In the middle is the picture histogram, and at the bottom is the three-label segmentation. (b) The top shows the bias-corrected MRI slice, the centre shows the related histogram, and the bottom shows three-label segmentation.

### 3.4. Implementation

Storing path and listing number of dataset

```
main_dir = "/content/drive/MyDrive/Sem 7/MinorProject/ppmi-org-data"
PD_flair_dir = "/content/drive/MyDrive/Sem 7/MinorProject/ppmi-org-data/PD/flair"
PD_nonflair_dir = "/content/drive/MyDrive/Sem 7/MinorProject/ppmi-org-data/PD/nonflair/"

HC_flair_dir = "/content/drive/MyDrive/Sem 7/MinorProject/ppmi-org-data/HC/flair/"
HC_nonflair_dir = "/content/drive/MyDrive/Sem 7/MinorProject/ppmi-org-data/HC/nonflair/"

PD_flair_dir_list = os.listdir(PD_flair_dir)
PD_nonflair_dir_list = os.listdir(PD_nonflair_dir)
HC_flair_dir_list = os.listdir(HC_flair_dir)
HC_nonflair_dir_list = os.listdir(HC_nonflair_dir)

print(len(PD_flair_dir_list))
print(len(PD_nonflair_dir_list))
print(len(HC_flair_dir_list))
print(len(HC_nonflair_dir_list))

99
124
34
43
```

Exploring 3D image

**Exploring the layers using widger-(interactive)**

```
import ipywidgets
# Define a function to visualize the data
def explore_3dimage(layer):
    plt.figure(figsize=(10, 5))
    channel = 0
    plt.imshow(nii_mask[:, :, layer, channel], cmap='gray');
    plt.title('Explore Layers of Brain MRI', fontsize=20)
    plt.axis('off')
    return layer

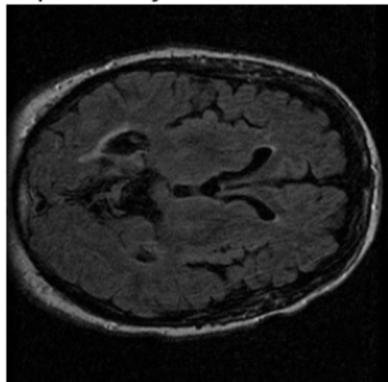
ipywidgets.interact(explore_3dimage, layer=(0, nii_mask.shape[2] - 1));
```



layer

15

Explore Layers of Brain MRI



15

## Preprocessing steps

Applying preprocessing techniques on a random image from the directory

Applied Otsu Thresholding and normalised image

```
[ ] PD_flair_dir1 = "/content/drive/MyDrive/Sem 7/MinorProject/ppmi-org-data/MadeByMe"
PD_flair_dir1_list = os.listdir(PD_flair_dir1)
import tensorflow as tf
import matplotlib.pyplot as plt
import cv2
import numpy as np
import random

randomnum = random.randint(0,98)

def preprocess(img):
    img = cv2.cvtColor(np.uint8(img), cv2.COLOR_RGB2GRAY)
    img2 = img
    img2 = cv2.equalizeHist(img2.astype(np.uint8))
    ret, img2 = cv2.threshold(img2, 0, 255, cv2.THRESH_OTSU)
    finalimg = cv2.subtract(img2,img)
    ret, finalimg = cv2.threshold(finalimg, 0, 255, cv2.THRESH_OTSU)
    finalimg = img2 + cv2.subtract(img,finalimg)
    #normalised_img = np.array(cv2.cvtColor(finalimg, cv2.COLOR_GRAY2RGB))/255.0
    return np.array(cv2.cvtColor(finalimg, cv2.COLOR_GRAY2RGB))/255.0

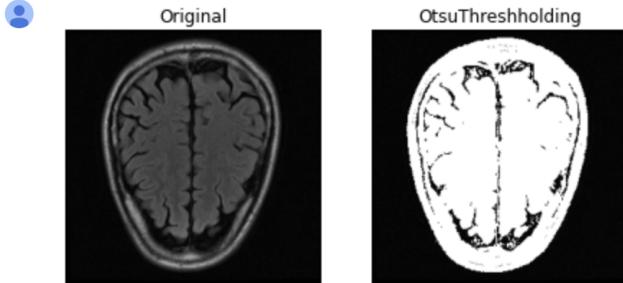
fig, ax = plt.subplots(1,2,figsize=(7,7))
image = tf.keras.preprocessing.image.load_img(PD_flair_dir1+'/'+PD_flair_dir1_list[randomnum])

ax[0].axis("off")
ax[0].title.set_text("Original")

ax[0].axis("off")
ax[0].title.set_text("Original")
ax[0].imshow(image)

img = preprocess(image)
ax[1].axis("off")
ax[1].title.set_text("OtsuThresholding")
ax[1].imshow(img)

plt.show()
```



#### Cropping the image to center

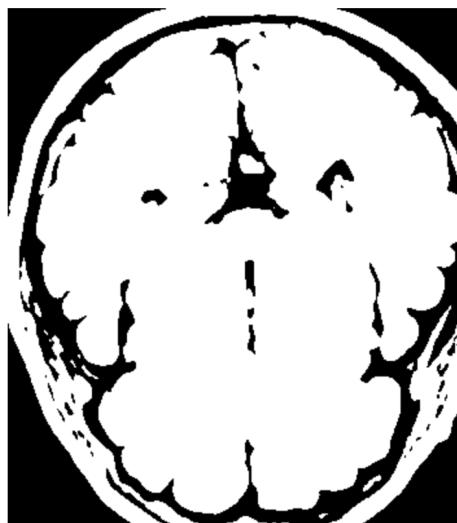
```
from google.colab.patches import cv2_imshow
image = cv2.imread('/content/drive/MyDrive/Sem 7/MinorProject/ppmi-org-data/MadeByMe/PD_flair_fPD(60.jpg')
#image = cv2.imread(PD_flair_dir1+'/' +PD_flair_dir1_list[randomnum])

#cv2_imshow(image)

# convert the image to grayscale and blur it slightly
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
blurred = cv2.GaussianBlur(gray, (7, 7), 0)

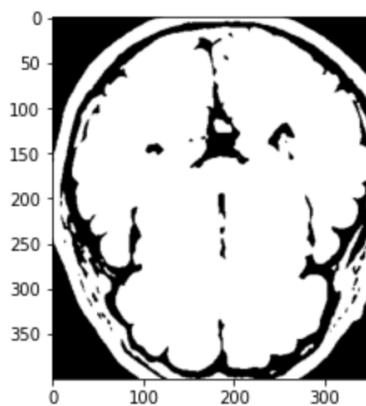
width, height = blurred.shape[1], blurred.shape[0]
dim1, dim2 = 350,400
crop_width = dim1 if dim1<blurred.shape[1] else blurred.shape[1]
crop_height = dim2 if dim2<blurred.shape[0] else blurred.shape[0]
mid_x, mid_y = int(width/2)-15, int(height/2)
cw2, ch2 = int(crop_width/2), int(crop_height/2)
crop_img = blurred[mid_y-ch2:mid_y+ch2, mid_x-cw2:mid_x+cw2]
cv2_imshow(crop_img)

[T, threshInv] = cv2.threshold(crop_img, np.mean(blurred), 255, cv2.THRESH_BINARY)
cv2_imshow(threshInv)
#cv2.threshold(img, np.mean(img), 255, cv2.THRESH_BINARY_INV)
```



```
binary_mask = crop_img > T

fig, ax = plt.subplots()
plt.imshow(binary_mask, cmap='gray')
plt.show()
```



## Segmentation steps

### Segmentation - Largest Connected component

```
[ ] ret, thresh_custom = cv2.threshold(crop_img,0,255, cv2.THRESH_BINARY+ cv2.THRESH_OTSU)

[ ] temp = crop_img
plt.imshow(crop_img, cmap = 'gray')
_,binarized = cv2.threshold(crop_img, 0 , 255, cv2.THRESH_BINARY + cv2.THRESH_OTSU)

foreground_value = 255
#mask = np.zeros((400,300), np.uint8(binarized == foreground_value))
mask = np.uint8(binarized == foreground_value)
mask_inv = cv2.bitwise_not(mask)
labels,stats = cv2.connectedComponentsWithStats(mask,4)[1:3]
largest_label = 1+ np.argmax(stats[1:, cv2.CC_STAT_AREA])
llc = np.zeros_like(binarized)
llc[labels == largest_label] = foreground_value

plt.imshow(binarized, cmap= 'gray')
fig,ax = plt.subplots(ncols=1,nrows=1,figsize = (5,5))
plt.imshow(llc, cmap='gray')
```

```
#mathematical morphology
kernel = np.ones((15,15), np.uint8)
llc_closing_image = cv2.morphologyEx(llc, cv2.MORPH_CLOSE, kernel)
print(kernel.shape, "kernel-value")
fig,ax = plt.subplots(ncols=1,nrows=1,figsize = (5,5))
plt.imshow(llc_closing_image)
```

```
(15, 15) kernel-value
<matplotlib.image.AxesImage at 0x7fe8da056e90>
```



### Skull-Stripping

```
[ ] skull_stripped_image = cv2.bitwise_and(crop_img,crop_img, mask = llc_closing_image)
brain_pixels = skull_stripped_image[llc_closing_image == foreground_value]
#print(brain_pixels, "skull-stripping brain pixels value")
#plt.imshow(brain_pixels)

[ ] #Adapting the data to k-means
kmeans_input = np.float32(brain_pixels.reshape(brain_pixels.shape[0],brain_pixels.ndim))
print(kmeans_input, "k-means value")

[[37.]
[37.]
[37.]
...
[43.]
[40.]
[36.]] k-means value

[ ] #K-means parameters
epsilon = 0.01
iterations = 10
clusters = 3
repetition = 10
criteria = (cv2.TERM_CRITERIA_EPS + cv2.TERM_CRITERIA_MAX_ITER, iterations, epsilon)
flags = cv2.KMEANS_RANDOM_CENTERS
print(criteria,"criteria Kmeans parameters")

(3, 10, 0.01) criteria Kmeans parameters

[ ] #K-means parameters
epsilon = 0.01
iterations = 10
clusters = 3
repetition = 10
criteria = (cv2.TERM_CRITERIA_EPS + cv2.TERM_CRITERIA_MAX_ITER, iterations, epsilon)
flags = cv2.KMEANS_RANDOM_CENTERS
print(criteria,"criteria Kmeans parameters")

(3, 10, 0.01) criteria Kmeans parameters

[ ] #K-means Segmentation
_,labels,centers = cv2.kmeans(kmeans_input, clusters, None, criteria, repetition, flags)
print(labels.shape,"K-means Segmentation")
#plt.imshow(labels)

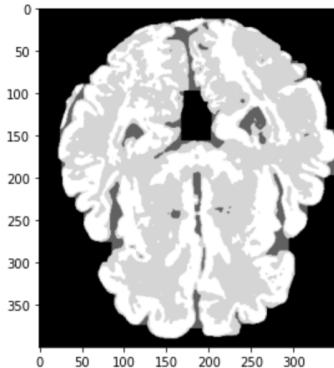
(90478, 1) K-means Segmentation

[ ] #adapting the labels
labels = labels.flatten('F')
for x in range(clusters):
    labels[labels == x] = centers[x]
```

## Segmented Image

```
▶ segmented_image = np.zeros_like(llc_closing_image)
segmented_image[llc_closing_image==foreground_value] = labels
print(segmented_image.shape,"Segmented image value")
print(labels.shape)
#ShowImage('Brain with skull XXX', segmented_image, 'gray')
fig,ax = plt.subplots(ncols=1,nrows=1,figsize = (5,5))
plt.imshow(segmented_image, cmap = 'gray')

(400, 350) Segmented image value
(90478,)
<matplotlib.image.AxesImage at 0x7fe8db7fd9d0>
```



## Loading images

```
import os
import cv2
import tensorflow as tf
from IPython.display import SVG
import tensorflow.keras.layers as L
import tensorflow.keras.backend as K
from tensorflow.keras.models import Model
from tensorflow.keras.optimizers import RMSprop, Adam
from tensorflow.keras.applications import DenseNet121
import numpy as np
from keras.utils import np_utils
from sklearn.model_selection import train_test_split

def load_image(filename):
    if (os.path.isfile(filename) == False):
        print("File Don't Exists")
    image = cv2.imread(filename)
    image = np.array(image).reshape(512, 512, 3).astype('float32')
    return image

# def load_image(filename):
#     bits = tf.io.read_file(filename)
#     image = tf.image.decode_jpeg(bits, channels=3)
#     image = tf.cast(image, tf.float32)
#     image = tf.image.resize(image, (512,512))
#     # return image
#     return tf.reshape(image, [-1, 512, 512, 3])
```

## Creating dataset

```
PD_images = []
HC_images = []

count = 0
for img in os.listdir(PD_PATH):
    count+=1
    if count > 500:
        break
    image = load_image(PD_PATH + img)
    PD_images.append(image)
    # predictions = model.predict(img)
    # print(predictions)

count = 0
for img in os.listdir(HC_PATH):
    count+=1
    if count > 500:
        break
    image = load_image(HC_PATH+img)
    HC_images.append(image)

PD_y = np.array([1]*int(len(PD_images)))
HC_y = np.array([0]*int(len(HC_images)))

len(PD_y), len(HC_y)
```

## Splitting data

```
from sklearn.model_selection import train_test_split

train_X, test_X, train_y, test_y = \
    train_test_split(New_dataset_X, New_dataset_y, test_size=0.15, random_state=1010)
# train_test_split(dataset['X'], dataset['y'], test_size=0.15, random_state=1010)
```

## Models we have used

```
# base_model = DenseNet121(input_shape = (512, 512, 3), include_top = False, weights = 'imagenet')
# MobileNet
base_model = keras.applications.mobilenet.MobileNet(input_shape = (512, 512, 3), include_top = False, weights = 'imagenet')
x = base_model.output
x = keras.layers.GlobalAveragePooling2D()(x)
# x = keras.layers.Dense(1024, activation='relu')(x)
# x = keras.layers.Dropout(0.2)(x)
# x = keras.layers.Dense(512,activation='relu')(x)
x = keras.layers.Dropout(0.2)(x)
preds = keras.layers.Dense(2,activation='softmax')(x)
model = keras.models.Model(inputs=base_model.input,outputs=preds)

model.compile(optimizer = Adam(lr=0.001), loss = 'binary_crossentropy', metrics = ['accuracy','mae'])
model.summary()
```

```
base_model = efn.EfficientNetB7(
    input_shape=(512, 512, 3),
    weights='imagenet',
    include_top=False)

x = base_model.output
x = keras.layers.GlobalAveragePooling2D()(x)
# x = keras.layers.Dense(1024, activation='relu')(x)
# x = keras.layers.Dropout(0.2)(x)
# x = keras.layers.Dense(512,activation='relu')(x)
x = keras.layers.Dropout(0.2)(x)
preds = keras.layers.Dense(2,activation='softmax')(x)
model = keras.models.Model(inputs=base_model.input,outputs=preds)

model.compile(optimizer = Adam(lr=0.001), loss = 'binary_crossentropy', metrics = ['accuracy','mae'])
model.summary()
```

## 4. Result Analysis

We have used the VGG net, MobileNet using the transfer learning concept to test the approach. Using the approach mentioned in the methodology section we found the below results on VGGnet and MobileNet.

Model	Accuracy
VGGNet	~75
MobileNet	90.82

For the VGGNet we have kept the configuration as learning rate to 0.001, training test data ratio was 85: 15, and in total we have used it to train on 1000 png images obtained after all the preprocessing step and segmentation steps. We have used the transfer learning concept and not included the top layers in VGGNet instead we have used the average pooling layer and dense layer as the output layer with the adam as the optimiser and binary-cross entropy as the loss function. The accuracy we obtained was ~75 percent.

For the MobileNet we have kept the configuration as learning rate to 0.001, training test data ratio was 85: 15, and in total we have used it to train on 1000 png images obtained after all the preprocessing steps and segmentation steps. We have used the transfer learning concept and not included the top layers in MobileNet instead we have used the global average pooling layer and dense layer as the output layer with the adam as the optimiser and binary-cross entropy as the loss function. The accuracy we obtained was 90.82%, we have kept the epoch value as 5.

## **F. Conclusion & Future Work**

### **Conclusion**

Three parts of brain tissue such as WM, This approach successfully segmented CSF and GM. Grayscaleing is used in the preprocessing step to make the segment process easier by converting the pixel value to 0 - 255. Our approach was able to discriminate between early-PD subjects and controls with an accuracy of 90.82%. In this paper, we have seen various uses of segmentation methods, why it's novel and how it significantly increases the accuracy of the model. Generally, segmentation segments the important part of the brain which include substantia nigra due to which the disease is caused.

### **Future Work**

For future works, we might include the data augmentation technique and try different new preprocessing techniques along with corresponding usable segmentation techniques. We might use the configuration on which our model can run efficiently, maybe on the supercomputer to make the complex calculation and high degree matrix multiplications run with less time rather than giving huge time for the same process.

### **References**

- [1]. Poewe W, Seppi K, Tanner CM, Halliday GM, Brundin P, Volkmann J, Schrag AE, Lang AE (2017) Parkinson disease. Nat Rev Dis Primers 3:17013
- [2]. Sivaranjini S et al., "Deep learning based diagnosis of Parkinson's disease using convolutional neural network." Multimedia Tools and Applications (2019): 1-13.
- [3]. Rummanet et al., "Early detection of Parkinson's disease using image processing and artificial neural network." In 2018 Joint 7th International Conference on Informatics, Electronics & Vision (ICIEV) and 2018 2nd International Conference on Imaging, Vision & Pattern Recognition (icIVPR), pp. 256-261. IEEE, 2018.
- [4]. Ivana Despotović, Bart Goossens, and Wilfried Philips, "MRI Segmentation of the Human Brain: Challenges, Methods, and Applications", Hindawi, 2014.

- [5]. Shah et al., "Detection of Parkinson Disease in Brain MRI using Convolutional Neural Network." In 2018 24th International Conference on Automation and Computing (ICAC), pp. 1-6. IEEE, 2018.
- [6]. K. Akyol, "Comparing of deep neural networks and extreme learning machines based on growing and pruning approach," Expert Systems With Applications, vol. 140, doi.org/10.1016/j.eswa.2019.112875 , 2020
- [7]. S. Kaur, H. Aggarwal, and R. Rani, "Hyper-parameter optimization of deep learning model for prediction of Parkinson's disease," Machine Vision and Applications, doi.org/10.1007/s00138-020-01078-1, 2020.
- [8]. D. Rizvi, I. Nissar, S. Masood, M. Ahmed, and F. Ahmad, "An LSTM based Deep Learning model for voice-based detection of Parkinson's disease," IJAST, vo. 29, pp. 337-343, 2020.
- [9]. Tuncer, Turker, Dogan, Sengul, Acharya, and U. Rajendra, "Automated detection of Parkinson's disease using minimum average maximum tree and singular value decomposition method with vowels," Biocybernetics and Biomedical Engineering, vol. 40(1), pp. 211-220, 2020.
- [10]. S.S.Upadhyay, and A.N. Cheeran, "Discriminating Parkinson and healthy people using phonation and cepstral features of speech," Procedia Comput Sci., vol. 143, pp. 197–202, 2018.
- [11]. Peng,Bo, et.al."Computer aided analysis of cognitive disorder in patients with Parkinson using machine learning method with multilevel ROI- based features." Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), International Congress on. IEEE, 2016.
- [12]. Pahuja, Gunjan, and T. N. Nagabhushan. "A novel GA-ELM approach for Parkinson's disease detection using brain structural T1-weighted MRI data." Cognitive Computing and Information Processing (CCIP), 2016 Second International Conference on. IEEE, 2016.
- [13]. "Parkinson Disease Detection Using Deep Neural Networks" Shivangi, Anubhav Johri and Ashish Tripathi, 2019
- [14]. Unsupervised Method for 3D Brain Magnetic Resonance Image Segmentation