

# Brain MR Image Segmentation using RBF Neural Network



*Under the Supervision of*  
**Prof. (Dr.) Jamuna Kanta Sing**

*By -*

**Shruti Pathak**

**Reg. No.- 163613 of 2022 - 2023**

**Exam Roll No.- MCA244020**

**MCA- 2<sup>nd</sup> Year 2<sup>nd</sup> Semester**

**JADAVPUR UNIVERSITY**

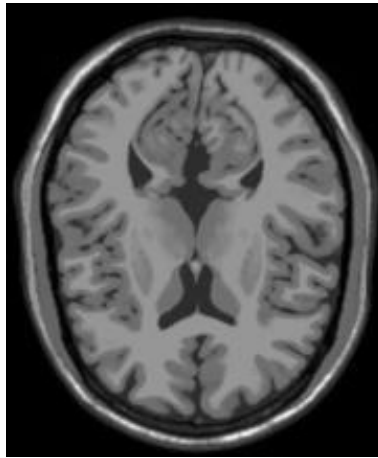
# Overview

- Introduction
- Proposed Method
- Experiments & Results
- Conclusion
- References

# Introduction

- The **brain**, as the central nervous system controlling speech, memory, cognition, and movement, faces increasing threats from diseases due to stress, lifestyle, injuries, and aging.
- **Magnetic Resonance Imaging (MRI)** is a cutting-edge, non-invasive technique for precisely visualizing the shape and function of brain tissues to diagnose brain illnesses.
- **Image segmentation** in brain MRIs is crucial for applications such as surgical planning, brain tissue classification, and tumor detection.
- Brain MRI segmentation involves dividing an MRI scan into distinct structures like **white matter**, **gray matter**, **CSF** (cerebrospinal fluid ), and regions such as the cortex, ventricles, or tumors.
- To make brain MRI segmentation more accurate when the structure varies, researchers use different methods. These range from basic image processing to more advanced machine learning and deep learning techniques.

# Introduction



WM



GM

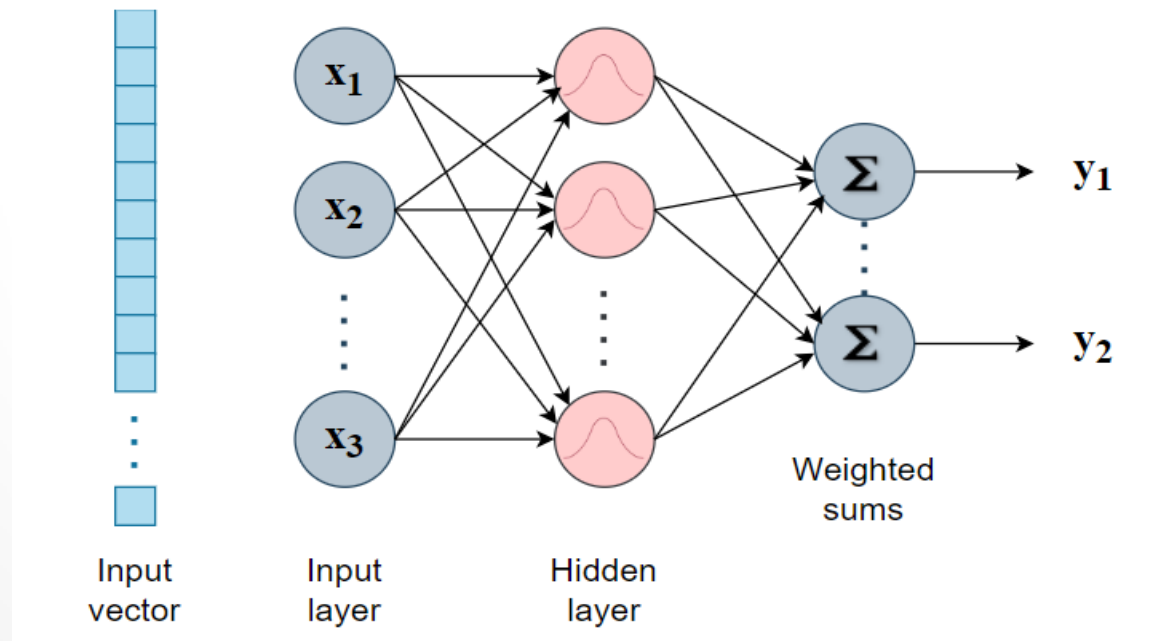


CSF



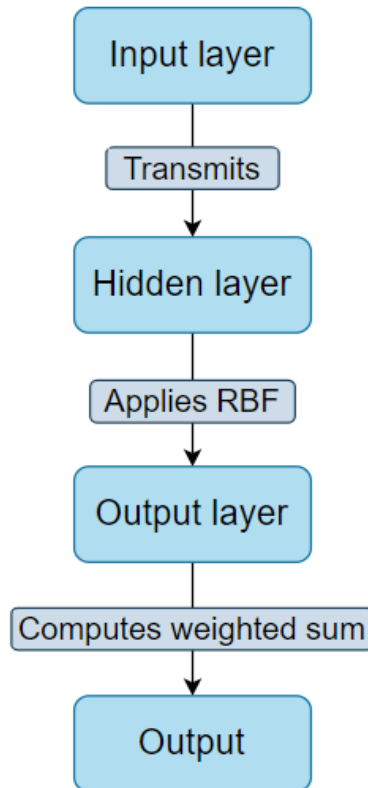
# Proposed Method

- In this project, a **Radial Basis Function Neural Network (RBFNN)** is employed specifically for the segmentation of brain MR images into white matter, grey matter, and cerebrospinal fluid (CSF) regions.
- A radial basis function (RBF) neural network is a type of artificial neural network that uses radial basis functions as activation functions.



# Proposed Method

- The network consists of **three layers**: an input layer, a hidden layer with radial basis functions, and an output layer.



# Proposed Method

The RBF network is a fully interconnected feed- forward network with **one hidden layer**. It can be mathematically described as follows:

$$Z_j = \frac{1}{M} \sum_{m=1}^M W_{mj} \Phi_m + b_j$$

Where,

$Z_j$  is the activation of the j-th output neuron,

$M$  is the number of hidden neurons,

$W_{mj}$  is the weight between m-th hidden and j-th output neuron and

$b_j$  is the bias term.

# Proposed Method

The output of hidden layer neuron is usually generated by a **Gaussian function** as follows:

$$\Phi_m = \exp \left[ \frac{- ||x - \mu||^2}{2\sigma_m^2} \right]$$

Here,

$x$  represents the input vector,

$\mu$  is the fixed center position and

$\sigma_m$  represents the fixed width of the m-th hidden layer neuron.



# Proposed Method

Radial basis function networks (RBFNs) work by comparing the input to known examples from the training data to classify it.

Here's a simplified explanation of how it works:

- ❑ RBFNs process input vectors through an input layer.
- ❑ RBF neurons in the hidden layer assess input proximity using Gaussian functions based on neuron centers.
- ❑ Hidden layer outputs are integrated in the output layer.
- ❑ Classification is achieved by computing weighted sums of hidden layer outputs.
- ❑ The final output represents the input's classification.

# Experiments

## Database:

The dataset of this study is downloaded from **BrainWeb**, which is acquired from the McConnell Brain Imaging Center of the Montreal Neurological Institute, McGill University .

This database contains a collection of realistic MRI data generated by an MRI simulator. It encompasses various parameters such as variations in slice thickness, noise levels, and intensity non-uniformity (INU).

As the “**Ground Truth**” , I utilize the discrete anatomical model, which assigns class labels (0=Background, 1=CSF, 2=Grey Matter, 3=White Matter etc..) to each voxel.

# Experiments

The dataset in my work consists of 11 T1-weighted MRI volumes on normal brain, each yielding 51 images of dimensions 181 pixels by 217 pixels, intentionally manipulated with varying noise and intensity non-uniformity (INU) levels , as meticulously delineated in the table below -

	Volume 1 (%)	Volume 2 (%)	Volume 3 (%)	Volume 4 (%)	Volume 5(%)	Volume 6 (%)	Volume 7 (%)	Volume 8 (%)	Volume 9 (%)	Volume 10 (%)	Volume 11 (%)
Noise	0	1	1	3	3	5	5	7	7	9	9
INU	0	20	40	20	40	20	40	20	40	20	40

# Experiments

- **Data Extraction:**

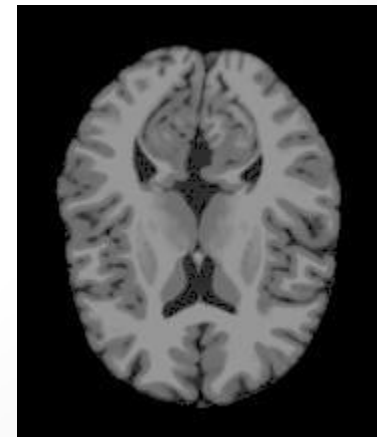
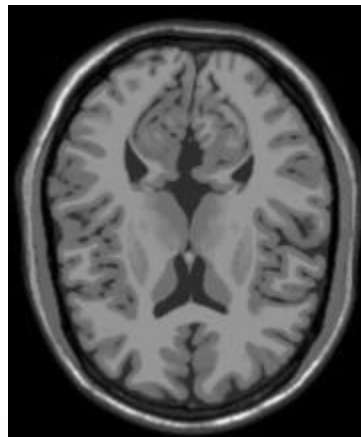
At first, I extracted the data from RAWB files into PGM files, and further transformed them into JPEG format for visualization and analysis.

- **Pre-Processing:**

Pre-processing brain MR images is fundamental for quality assurance in quantitative analysis, involving operations to enhance quality and remove non-brain tissues, laying the foundation for accurate segmentation.

- ☐ **Skull- stripping**

- ☐ **Normalization**

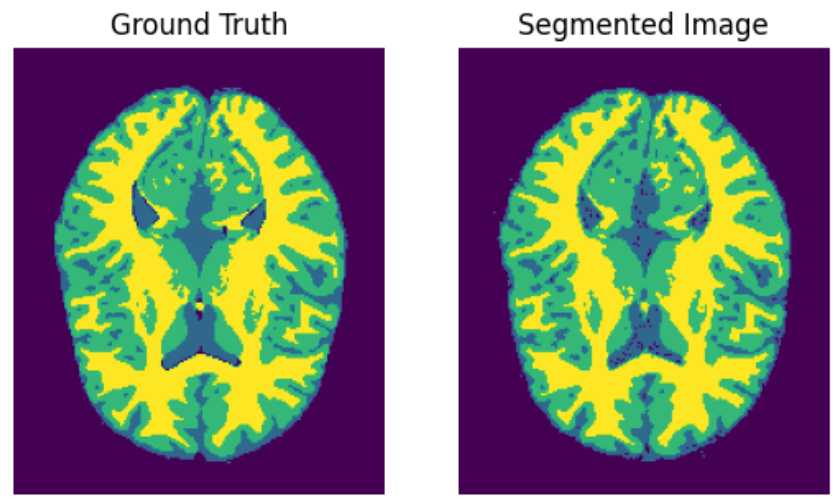


# Experiments

- **Segmentation:**

Then I use RBFNN for segmenting brain MRI volumes into white matter, grey matter, and cerebrospinal fluid (CSF), accommodating variations in noise and intensity non-uniformity (INU) percentages across multiple volumes, ensuring comprehensive analysis and reliable segmentation results across diverse imaging conditions.

- ☐ **Split the datasets**
- ☐ **Standardization**
- ☐ **Training**
- ☐ **Evaluation**



# Experiments

- **Evaluation Metrics:**

In this experiment, I utilized **Accuracy** , **F<sub>1</sub> Score** and **MSE** (Mean Squared Error) as '**loss**' for evaluation. To calculate accuracy and F<sub>1</sub> score, **Precision** and **Recall** must be computed first. Precision and Recall can be determined using the equations mentioned below:

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

Here,

**TP**- True Positive (Model has predicted Positive class whether it is actually positive)

**FP**- False Positive (Model has predicted Positive class whether it is actually negative) and

**FN**- False Negative (Model has predicted Negative class whether it is actually positive)

# Experiments

Accuracy and  $F_1$  Score can be computed using the equation mentioned below –

$$Accuracy = \frac{TP}{Total\ Number\ of\ Actual\ Positives}$$

$$F_1\ Score = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

Accuracy and  $F_1$  Score both lie between  $[0,1]$ , **higher the values** better Recognition has been done.

# Experiments

During model training, **Mean Squared Error** (MSE) served as the **loss function**. MSE calculates the average squared difference between predicted and actual values. Its range is [0.0, 1.0], with values **closer to 0** indicating better performance.

The formula for calculating MSE is as follows:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

Where:

$n$  is the number of samples

$y_i$  is the true label for sample  $i$

$\hat{y}_i$  is the predicted label for sample  $i$



# Results

Experiment Results with Different Data Partitioning Ratios –  
**Dataset- 70% Train, 15% Validation, 15% Test :-**

Volume (Noise, IHH)	Region	Precision	Recall	F1 score	Accuracy (Total)	Loss – 'MSE' (Total)
Volume 1 (0,0)	WM	0.9877	0.9461	0.9665	0.9650	0.0396
	GM	0.9204	0.9696	0.9444		
	CSF	0.8804	0.8818	0.8811		
Volume 2 (1,20)	WM	0.9654	0.9507	0.9580	0.9539	0.04901
	GM	0.9244	0.9385	0.9314		
	CSF	0.8469	0.8035	0.8246		
Volume 3 (1,40)	WM	0.9549	0.9069	0.9303	0.9362	0.0626
	GM	0.8816	0.9308	0.9055		
	CSF	0.8386	0.7067	0.7670		
Volume 4 (3,20)	WM	0.9747	0.9181	0.9455	0.9420	0.0584
	GM	0.8958	0.9433	0.9189		
	CSF	0.8212	0.7184	0.7664		
Volume 5 (3,40)	WM	0.9551	0.9046	0.9292	0.9330	0.0658
	GM	0.8753	0.9339	0.9037		
	CSF	0.8343	0.6664	0.7410		
Volume 6 (5,20)	WM	0.9312	0.9538	0.9424	0.9367	0.0638
	GM	0.9162	0.9060	0.9111		
	CSF	0.8327	0.6510	0.7307		
Volume 7 (5,40)	WM	0.9327	0.9118	0.9221	0.9252	0.0728
	GM	0.8759	0.9090	0.8921		
	CSF	0.8076	0.6234	0.7037		
Volume 8 (7,20)	WM	0.9093	0.9116	0.9104	0.9207	0.0764
	GM	0.8689	0.8785	0.8736		
	CSF	0.8034	0.6417	0.7135		
Volume 9 (7,40)	WM	0.8968	0.9114	0.9040	0.9133	0.0823
	GM	0.8570	0.8766	0.8667		
	CSF	0.8090	0.5571	0.6598		
Volume 10 (9,20)	WM	0.8751	0.8581	0.8665	0.8962	0.0933
	GM	0.8081	0.8458	0.8265		
	CSF	0.7494	0.5964	0.6642		
Volume 11 (9,40)	WM	0.8607	0.8679	0.8643	0.8915	0.0975
	GM	0.8090	0.8341	0.8214		
	CSF	0.7566	0.5273	0.6215		

# Results

**Dataset- 60% Train, 20% Validation, 20% Test :-**

Volume (Noise, IHH)	Region	Precision	Recall	F1 score	Accuracy (Total)	Loss – 'MSE' (Total)
Volume 1 (0,0)	WM	0.9822	0.9541	0.9680	0.9654	0.0458
	GM	0.9343	0.9561	0.9451		
	CSF	0.8539	0.8934	0.8732		
Volume 2 (1,20)	WM	0.9614	0.9492	0.9552	0.9547	0.0472
	GM	0.9226	0.9422	0.9323		
	CSF	0.8585	0.7986	0.8275		
Volume 3 (1,40)	WM	0.9496	0.9167	0.9329	0.9387	0.0595
	GM	0.8940	0.9257	0.9096		
	CSF	0.8247	0.7188	0.7681		
Volume 4 (3,20)	WM	0.9737	0.9202	0.9462	0.9443	0.0552
	GM	0.9023	0.9419	0.9216		
	CSF	0.8130	0.7320	0.7704		
Volume 5 (3,40)	WM	0.9420	0.9254	0.9336	0.9360	0.0619
	GM	0.8996	0.9144	0.9070		
	CSF	0.8107	0.6864	0.7434		
Volume 6 (5,20)	WM	0.9341	0.9514	0.9426	0.9391	0.0602
	GM	0.9154	0.9135	0.9145		
	CSF	0.8239	0.6661	0.7367		
Volume 7 (5,40)	WM	0.9267	0.9152	0.9209	0.9272	0.0694
	GM	0.8833	0.9044	0.8937		
	CSF	0.7990	0.6415	0.7116		
Volume 8 (7,20)	WM	0.9010	0.9221	0.9114	0.9232	0.0718
	GM	0.8830	0.8688	0.8759		
	CSF	0.7836	0.6708	0.7228		
Volume 9 (7,40)	WM	0.9026	0.9031	0.9028	0.9157	0.0779
	GM	0.8634	0.8768	0.8700		
	CSF	0.7775	0.5940	0.6735		
Volume 10 (9,20)	WM	0.8760	0.8567	0.8662	0.8988	0.0894
	GM	0.8168	0.8453	0.8308		
	CSF	0.7436	0.6007	0.6646		
Volume 11 (9,40)	WM	0.8626	0.8632	0.8629	0.8938	0.0936
	GM	0.8186	0.8291	0.8239		
	CSF	0.7361	0.5466	0.6274		

# Results

**Dataset- 50% Train, 25% Validation, 25% Test :-**

Volume (Noise, IHH)	Region	Precision	Recall	F1 score	Accuracy (Total)	Loss – 'MSE' (Total)
Volume 1 (0,0)	WM	0.9814	0.9565	0.9688	0.9659	0.0363
	GM	0.9338	0.9569	0.9452		
	CSF	0.8663	0.8686	0.8675		
Volume 2 (1,20)	WM	0.9654	0.9489	0.9571	0.9561	0.0445
	GM	0.9212	0.9426	0.9318		
	CSF	0.8526	0.8034	0.8273		
Volume 3 (1,40)	WM	0.9497	0.9239	0.9367	0.9414	0.0574
	GM	0.8979	0.9223	0.9099		
	CSF	0.8280	0.7188	0.7696		
Volume 4 (3,20)	WM	0.9708	0.9327	0.9514	0.9482	0.0518
	GM	0.9061	0.9466	0.9259		
	CSF	0.8349	0.7220	0.7744		
Volume 5 (3,40)	WM	0.9428	0.9312	0.9370	0.9392	0.0605
	GM	0.8979	0.9198	0.9087		
	CSF	0.8256	0.6834	0.7478		
Volume 6 (5,20)	WM	0.9491	0.9419	0.9455	0.9408	0.0587
	GM	0.9141	0.9123	0.9132		
	CSF	0.7926	0.6882	0.7367		
Volume 7 (5,40)	WM	0.9227	0.9283	0.9255	0.9300	0.0667
	GM	0.8923	0.8938	0.8930		
	CSF	0.7974	0.6444	0.7127		
Volume 8 (7,20)	WM	0.8971	0.9323	0.9144	0.9256	0.0697
	GM	0.8859	0.8641	0.8749		
	CSF	0.7925	0.6661	0.7238		
Volume 9 (7,40)	WM	0.9020	0.9132	0.9075	0.9189	0.0749
	GM	0.8708	0.8670	0.8689		
	CSF	0.7682	0.6102	0.6802		
Volume 10 (9,20)	WM	0.8553	0.9001	0.8771	0.9030	0.0864
	GM	0.8420	0.8106	0.8260		
	CSF	0.7487	0.5998	0.6661		
Volume 11 (9,40)	WM	0.8546	0.8872	0.8706	0.8980	0.0909
	GM	0.8268	0.8182	0.8225		
	CSF	0.7575	0.5328	0.6256		

# Results

**Dataset- 40% Train, 30% Validation, 30% Test :-**

Volume (Noise, IHH)	Region	Precision	Recall	F1 score	Accuracy (Total)	Loss – ‘MSE’ (Total)
Volume 1 (0,0)	WM	0.9907	0.9375	0.9633	0.9636	0.0427
	GM	0.9179	0.9554	0.9363		
	CSF	0.8348	0.8974	0.8650		
Volume 2 (1,20)	WM	0.9570	0.9625	0.9597	0.9581	0.0426
	GM	0.9334	0.9297	0.9315		
	CSF	0.8515	0.8075	0.8289		
Volume 3 (1,40)	WM	0.9504	0.9299	0.9400	0.9442	0.0545
	GM	0.8995	0.9224	0.9108		
	CSF	0.8353	0.7197	0.7732		
Volume 4 (3,20)	WM	0.9718	0.9357	0.9534	0.9494	0.0509
	GM	0.9109	0.9363	0.9234		
	CSF	0.8128	0.7381	0.7737		
Volume 5 (3,40)	WM	0.9467	0.9339	0.9402	0.9420	0.0568
	GM	0.8961	0.9247	0.9102		
	CSF	0.8514	0.6616	0.7446		
Volume 6 (5,20)	WM	0.9498	0.9488	0.9493	0.9446	0.0553
	GM	0.9135	0.9217	0.9176		
	CSF	0.8250	0.6702	0.7396		
Volume 7 (5,40)	WM	0.9077	0.9467	0.9268	0.9317	0.0655
	GM	0.9011	0.8773	0.8890		
	CSF	0.8199	0.6271	0.7107		
Volume 8 (7,20)	WM	0.8981	0.9390	0.9181	0.9284	0.0718
	GM	0.8875	0.8626	0.8748		
	CSF	0.7999	0.6616	0.7242		
Volume 9 (7,40)	WM	0.9030	0.9202	0.9115	0.9221	0.0719
	GM	0.8706	0.8681	0.8693		
	CSF	0.7841	0.5987	0.6789		
Volume 10 (9,20)	WM	0.8786	0.8856	0.8821	0.9074	0.0834
	GM	0.8297	0.8350	0.8323		
	CSF	0.7465	0.6041	0.6678		
Volume 11 (9,40)	WM	0.8714	0.8822	0.8768	0.9028	0.0862
	GM	0.8237	0.8302	0.8269		
	CSF	0.7525	0.5454	0.6325		

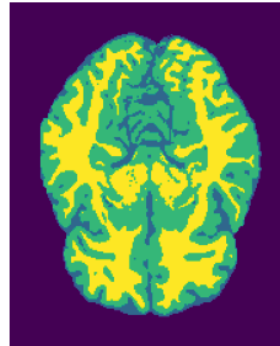
# Results

**Dataset- 30% Train, 35% Validation, 35% Test :-**

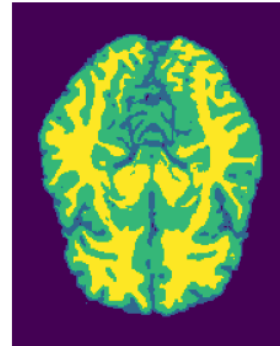
Volume (Noise, IHH)	Region	Precision	Recall	F1 score	Accuracy (Total)	Loss – ‘MSE’ (Total)
Volume 1 (0,0)	WM	0.9784	0.9633	0.9708	0.9679	0.0363
	GM	0.9357	0.9575	0.9465		
	CSF	0.8785	0.8714	0.8749		
Volume 2 (1,20)	WM	0.9680	0.9495	0.9587	0.9575	0.0443
	GM	0.9204	0.9445	0.9323		
	CSF	0.8619	0.7975	0.8284		
Volume 3 (1,40)	WM	0.9491	0.9235	0.9361	0.9421	0.0573
	GM	0.8926	0.9234	0.9077		
	CSF	0.8341	0.7240	0.7752		
Volume 4 (3,20)	WM	0.9738	0.9334	0.9532	0.9502	0.0501
	GM	0.9042	0.9511	0.9270		
	CSF	0.8490	0.7203	0.7794		
Volume 5 (3,40)	WM	0.9390	0.9412	0.9401	0.9416	0.0577
	GM	0.9030	0.9159	0.9094		
	CSF	0.8441	0.6777	0.7518		
Volume 6 (5,20)	WM	0.9440	0.9520	0.9479	0.9432	0.0571
	GM	0.9171	0.9140	0.9155		
	CSF	0.8098	0.6886	0.7443		
Volume 7 (5,40)	WM	0.9255	0.9306	0.9280	0.9320	0.0652
	GM	0.8930	0.8934	0.8932		
	CSF	0.7933	0.6576	0.7191		
Volume 8 (7,20)	WM	0.8847	0.9448	0.9138	0.9247	0.0728
	GM	0.8997	0.8326	0.8649		
	CSF	0.7680	0.6901	0.7270		
Volume 9 (7,40)	WM	0.9230	0.8897	0.9061	0.9192	0.0784
	GM	0.8535	0.8811	0.8671		
	CSF	0.7507	0.6273	0.6835		
Volume 10 (9,20)	WM	0.8608	0.9022	0.8810	0.9058	0.0846
	GM	0.8389	0.8165	0.8276		
	CSF	0.7607	0.5944	0.6674		
Volume 11 (9,40)	WM	0.8761	0.8719	0.8740	0.9010	0.0884
	GM	0.8207	0.8303	0.8254		
	CSF	0.7271	0.5776	0.6438		

# Results

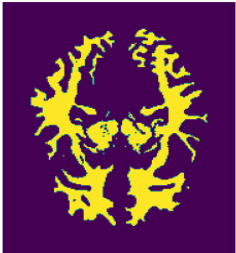
Ground Truth



Segmented Image



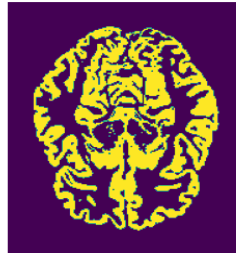
Original White Matter



Predicted White Matter



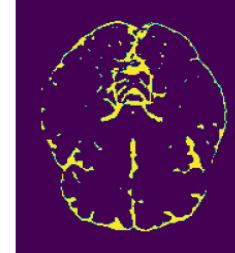
Original Grey Matter



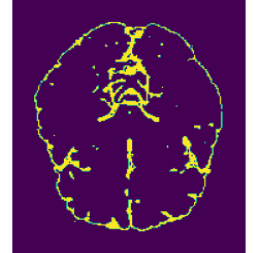
Predicted Grey Matter



Original CSF



Predicted CSF

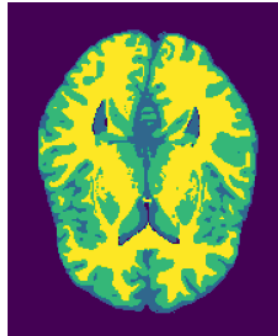


The original and the predicted masks of a random slice  
(slice- 62) from volume – 1(0,0)

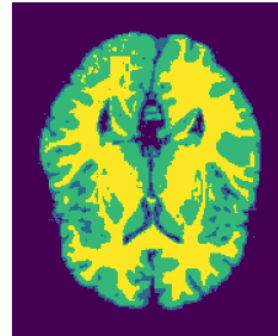


# Results

Ground Truth



Segmented Image



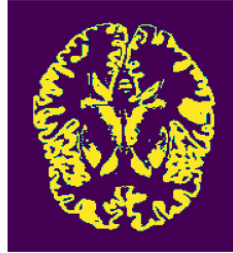
Original White Matter



Predicted White Matter



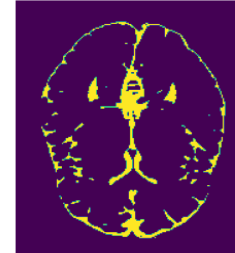
Original Grey Matter



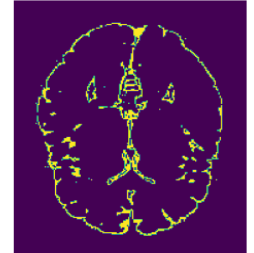
Predicted Grey Matter



Original CSF



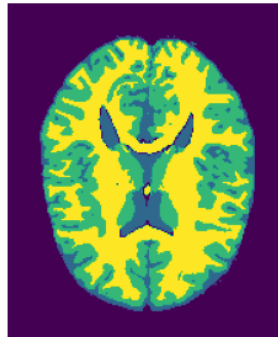
Predicted CSF



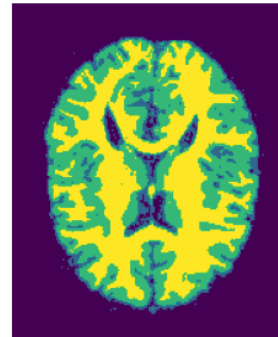
The original and the predicted masks of a random slice  
(slice- 73) from volume – 3(1,40)

# Results

Ground Truth



Segmented Image



Original White Matter



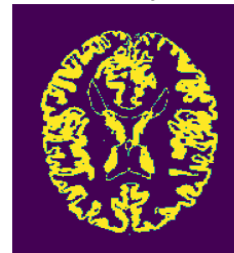
Predicted White Matter



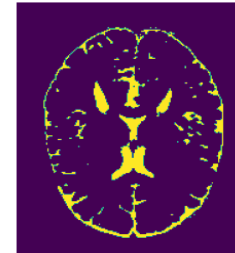
Original Grey Matter



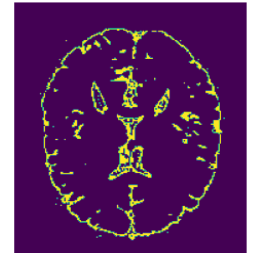
Predicted Grey Matter



Original CSF



Predicted CSF

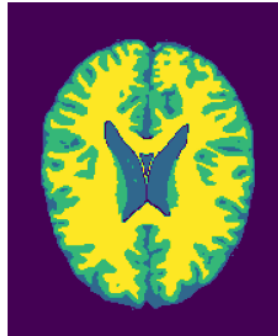


The original and the predicted masks of a random slice  
(slice- 85) from volume – 4(3,20)

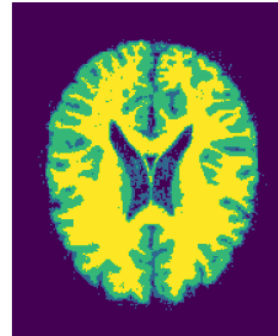


# Results

Ground Truth



Segmented Image



Original White Matter



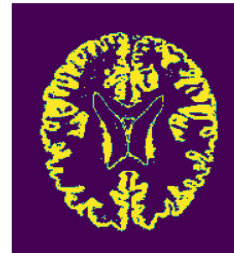
Predicted White Matter



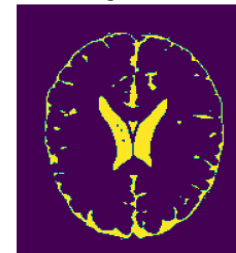
Original Grey Matter



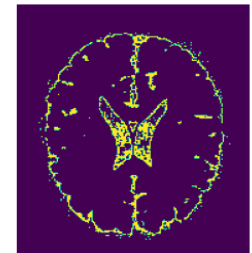
Predicted Grey Matter



Original CSF



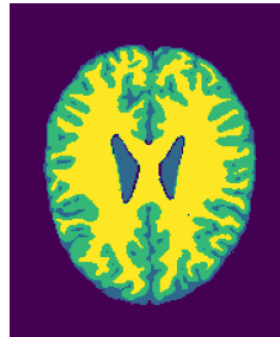
Predicted CSF



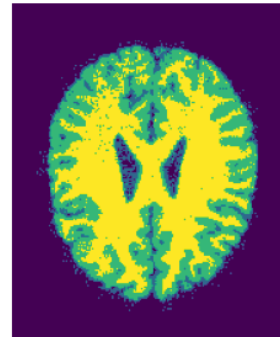
**The original and the predicted masks of a random slice  
(slice- 92) from volume – 6(5,20)**

# Results

Ground Truth



Segmented Image



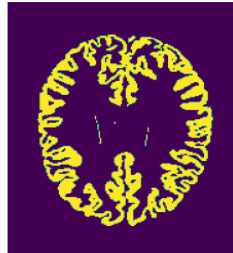
Original White Matter



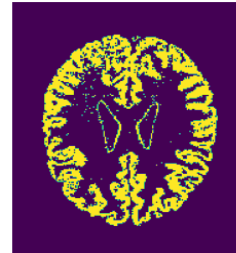
Predicted White Matter



Original Grey Matter



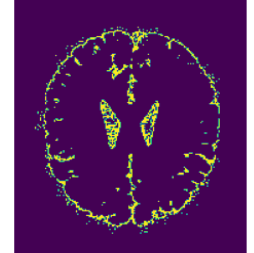
Predicted Grey Matter



Original CSF



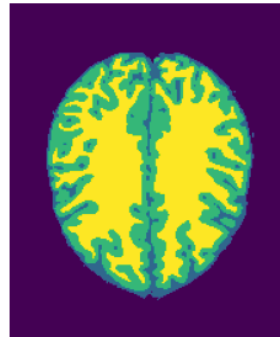
Predicted CSF



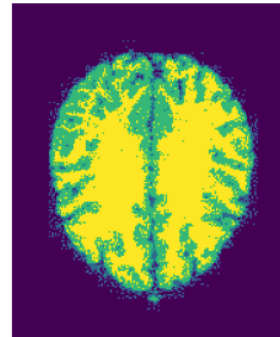
The original and the predicted masks of a random slice  
(slice- 98) from volume – 9(7,40)

# Results

Ground Truth



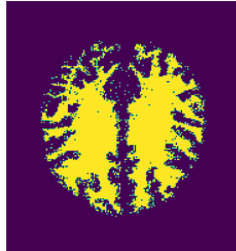
Segmented Image



Original White Matter



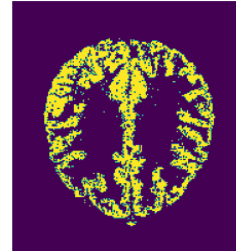
Predicted White Matter



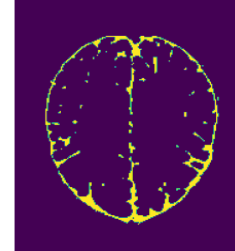
Original Grey Matter



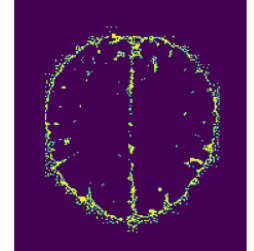
Predicted Grey Matter



Original CSF



Predicted CSF



The original and the predicted masks of a random slice  
(slice- 108) from volume – 11(9,40)

# Conclusion

- Medical image segmentation is vital for visualization, diagnosis, and treatment planning in clinical settings.
- Ongoing advancements in medical models lead to new segmentation challenges and the need for new techniques.
- This project highlights the effectiveness of Radial Basis Function Neural Network (RBFNN) for accurate brain MRI segmentation, delineating structures like CSF, grey matter, and white matter with promising outcomes.
- RBFNN's ability to identify patterns enables precise segmentation, while future research may explore other ML and Deep Learning methods like CNNs.
- It is important to acknowledge that the proposed algorithm is not without challenges. The increased complexity and computational requirements may lead to longer processing times. Despite computational challenges, RBFNN advancements offer hope for improved efficiency.
- Ultimately, this study demonstrates the effectiveness of RBFNN in brain MRI segmentation, contributing to the understanding of neuroanatomy and supporting clinical diagnosis and therapy planning for neurological illnesses.

# References

- [Sing, J. K., D. K. Basu, M. Nasipuri, and M. Kundu. "Self-adaptive RBF neural network-based segmentation of medical images of the brain." In Proceedings of 2005 International Conference on Intelligent Sensing and Information Processing, 2005., pp. 447-452. IEEE, 2005.](#)
- [Sing, J. K., Basu, D. K., Nasipuri, M., & Kundu, M. \(2005\). Segmentation of MR images of the human brain using fuzzy adaptive radial basis function neural network. In Pattern Recognition and Machine Intelligence: First International Conference, PReMI 2005, Kolkata, India, December 20-22, 2005. Proceedings 1 \(pp. 364-368\). Springer Berlin Heidelberg.](#)
- [Hyunseok Seo, Masoud Badiei Khuzani, Varun Vasudevan, Charles Huang, Hongyi Ren, Ruoxiu Xiao, Xiao Jia, Lei Xing1, "Machine Learning Techniques for Biomedical Image Segmentation: An Overview of Technical Aspects and Introduction to State-of-Art Applications", Med Phys. 2020 Jun; 47\(5\): e148–e167.](#)
- [Slabaugh, Greg, Quynh Dinh, and Gozde Unal. "A variational approach to the evolution of radial basis functions for image segmentation." In 2007 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1-8. IEEE, 2007.](#)
- [https://en.wikipedia.org/wiki/Radial\\_basis\\_function\\_network](https://en.wikipedia.org/wiki/Radial_basis_function_network)
- <https://www.educative.io/answers/what-are-radial-basis-function-neural-networks>

Thank You