

Radiology and Histopathology Induced Computer Vision for Neurological Disorder Detection

Thesis submitted by
Arijit De

Doctor of Philosophy (Ph.D.)
in
Engineering

Dept. of Electronics &
Telecommunication Engineering,
Faculty Council of Engineering & Technology
Jadavpur University, Kolkata
India
2024

**JADAVPUR UNIVERSITY
KOLKATA –700032, INDIA**

INDEX NO.: 254/20/E

1. Title of the Thesis:

“Radiology and Histopathology Induced Computer Vision for Neurological Disorder Detection”

2. Name, Designation & Institution of the Supervisor:

Dr. Ananda Shankar Chowdhury

Professor

Department of Electronics and Telecommunication Engineering

Jadavpur University, Kolkata, India

3. List of publication:

(In International Journal)

1. **A. De**, A. S. Chowdhury: “DTI based Alzheimer’s Disease Classification with Rank Modulated Fusion of CNNs and Random Forest”, *Expert Syst. Appl.* 169 (2021), 114338.
2. **A. De**, A. Santra, M. Tiwari, A. S. Chowdhury: “Predicting Genetic Markers for Brain Tumors Using a Composite Loss”, *IEEE/ACM Transactions on Computational Biology and Bioinformatics*. (Communicated)

(In International Conference)

1. **A. De**, M. Tiwari, E. Grisan, A. S. Chowdhury: “A Deep Graph Cut Model for 3D Brain Tumor Segmentation”, *44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, Glasgow, Scotland, United Kingdom (2022), pp. 2105-2109.

2. **A. De**, R. Mhatre, M. Tiwari, A. S. Chowdhury: “Brain Tumor Classification from Radiology and Histopathology using Deep Features and Graph Convolutional Network”, *26th International Conference on Pattern Recognition (ICPR)*, Montreal, QC, Canada (2022), pp. 4420-4426.
3. **A. De**, M. Tiwari, A. S. Chowdhury: “3D Hippocampus Segmentation Using a Hog Based Loss Function with Majority Pooling”, *IEEE International Conference on Image Processing (ICIP)*, Kuala Lumpur, Malaysia, (2023), pp. 2260-2264.
4. **A. De**, A. S. Chowdhury: “Shape induced Multi-Class Deep Graph Cut for Hippocampus Subfield Segmentation”, *27th International Conference on Pattern Recognition (ICPR)*, Kolkata, India, (2024) (accepted).

4. List of Patents: NIL

5. List of Presentations in National/International Conferences:

1. Presented the paper entitled “A Deep Graph Cut Model for 3D Brain Tumor Segmentation”, *44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, 11-15 July 2022 Glasgow, Scotland, United Kingdom.
2. Presented the paper entitled “Brain Tumor Classification from Radiology and Histopathology using Deep Features and Graph Convolutional Network”, *26th International Conference on Pattern Recognition (ICPR)*, 21-25 August 2022, Montreal, QC, Canada.
3. Presented the paper entitled “3D Hippocampus Segmentation Using a Hog Based Loss Function with Majority Pooling”, *IEEE International Conference on Image Processing (ICIP)*, 08-11 October 2023, Kuala Lumpur, Malaysia.
4. To present the paper entitled “Shape induced Multi-Class Deep Graph Cut for Hippocampus Subfield Segmentation”, *27th International Conference on Pattern Recognition (ICPR)*, 01-05 December 2024, Kolkata, India.

STATEMENT OF ORIGINALITY

I, *Arijit De*, registered on *03/02/2020* do hereby declare that this thesis entitled "*Radiology and Histopathology Induced Computer Vision for Neurological Disorder Detection*" contains literature survey and original research work done by the undersigned candidate as part of Doctoral studies.

All the information in this thesis have been obtained and presented in accordance with existing academic rules and ethical conduct. I declare that, as required by these rules and conduct, I have fully cited and referred all materials and results that are not original to this work.

I also declare that I have checked this thesis as per the "Policy on Anti Plagiarism, Jadavpur University, 2019", and the level of similarity as checked by iThenticate software is 8%.

Arijit De

Signature of Candidate

Date: 19/09/2024

Certified by Supervisor(s):

Ananda Shankar Chowdhury

Signature of the Supervisor

Dr. Ananda Shankar Chowdhury

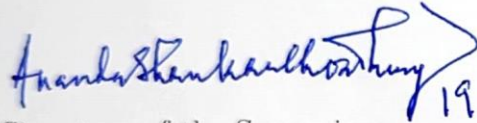
Professor, Dept. of ETCE

Jadavpur University, Kolkata - 700032

19-09-2024
ANANDA SHANKAR CHOWDHURY
Professor
Electronics and Telecommunication Engineering
Jadavpur University, Kolkata-32

CERTIFICATE FROM THE SUPERVISOR/S

This is to certify that the thesis entitled "*Radiology and Histopathology Induced Computer Vision for Neurological Disorder Detection*" submitted by *Shri Arijit De*, who got his name registered on *03/02/2020 [D-7/E/79/20]* for the award of Ph.D (Engg.) degree of Jadavpur University is absolutely based upon his own work under the supervision of *Prof. Ananda Shankar Chowdhury* and that neither his thesis nor any part of this thesis has been submitted for any degree/diploma or any other academic award anywhere before.



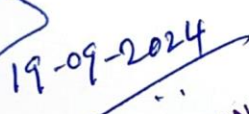
Signature of the Supervisor

Dr. Ananda Shankar Chowdhury

Professor, Dept. of ETCE

Jadavpur University, Kolkata - 700032

Date: 19/09/2024


ANANDA SHANKAR CHOWDHURY
Professor
Electronics and Telecommunication Engineering
Jadavpur University, Kolkata-32

Abstract

Brain abnormalities, including tumors, cancers, and neurodegenerative diseases, are rising globally due to factors like pollution, unhealthy lifestyles, and stress. Neurodegenerative diseases such as Alzheimer’s and Epilepsy are particularly challenging as they remain incurable, and their diagnosis is often time-consuming. The complexity of the brain makes identifying the causes of these disorders difficult, hindering effective treatment strategies. Consequently, neurologists and neuroscientists are increasingly relying on computer scientists to enhance diagnostic procedures and facilitate faster, more effective treatment. This thesis focuses on leveraging computer vision techniques to tackle real-world problems related to the diagnosis, treatment, and primary care of brain disorders, specifically Alzheimer’s disease and brain tumors. We first discuss about the works related to brain tumors. We first developed a 3D brain tumor segmentation model. Accurate segmentation is vital for diagnosis and surgery. The deep graph cut (DGC) model that we proposed combines a 3D UNet with graph cut algorithms, refining the energy function using UNet probability maps. This method outperformed existing approaches and the model’s robustness and accuracy make it valuable for clinical applications. Brain tumor surgery can benefit from segmentation, but treatment planning requires tumor classification. For this task, we did another work and developed a coarse-to-fine approach using radiology and histopathology data. A 3D CNN model first detects glioblastoma from MRI images. Then, 2D CNN features are extracted from Whole Slide Images (WSI) at two magnification levels. A Graph Convolutional Network (GCN) classifies non-glioblastoma cases. This method achieved a balanced accuracy of 91.4% on the CPM-RadPath2020 dataset, demonstrating enhanced classification accuracy and resilience. Further investigation focused on predicting genetic markers for brain tumors, essential for treatment planning. We developed an algorithm to predict five glioma-causing biomarkers (IDH, ATRX, MGMT, 1p/19q codeletion, and TERT) using WSI and genetic data. A composite loss function captures individual, pairwise, and group behaviors of the biomarkers. This approach achieved state-of-the-art pre-

diction performance on a new benchmark dataset, providing a comprehensive prognosis and treatment strategy for glioma patients. We then studied neurodegenerative diseases and solved certain difficult problems to improve therapy and reduce neuro-radiologist's and neurosurgeon's workload. Our first approach classifies Alzheimer's into four stages using Diffusion Tensor Imaging (DTI) data. The leading cause of dementia is Alzheimer's disease (AD), causing memory loss and cognitive decline. This work uses 3D DTI scans of a patient to classify the stage of AD it belongs to. The stages are Early Mild Cognitive Impairment (EMCI), Late Mild Cognitive Impairment (LMCI), AD and Normal Control (CN), i.e., healthy patient. This is the first work that attempts classification into four classes simultaneously. Separate Convolutional Neural Networks are trained on DTI-derived values, and a Random Forest Classifier (RFC) is trained on average values for each brain region. The combined method achieved a 92.6% classification accuracy on the ADNI database, demonstrating the efficacy of this approach. Next, we focused on hippocampus segmentation, crucial for diagnosing neurodegenerative diseases. The proposed method uses a 3D Attention UNet combined with a Histogram of Oriented Gradients (HOG) based loss function, capturing shape and structural details. Validated on public datasets, the method outperformed several state-of-the-art techniques, demonstrating its effectiveness for accurate 3D hippocampus segmentation. We further refined this work to segment subregions within the hippocampus. We present a novel deep graph cut approach for hippocampus subfield segmentation, crucial for diagnosing neurodegenerative diseases. The method incorporates deep learned shape information into the energy function of a graph cut and uses a modified α - β swap technique to improve segmentation accuracy and execution time. The proposed method outperformed several state-of-the-art techniques, achieving superior segmentation results. Overall, this thesis demonstrates the potential of computer vision techniques to significantly improve the diagnosis and treatment of brain disorders, offering valuable tools for clinicians and researchers.

Acknowledgment

*“What you get by achieving your goals is not as important
as what you become by achieving your goals”.*

— Zig Ziglar

The satisfaction that accompanies the successful completion of PhD would be incomplete without the mention of the people who made it possible and whose constant guidance and encouragement crown all the efforts with success.

There is no proper way to acknowledge everyone whose consistent effort, assistance and guidance have played a key role in this journey and finally leading to this dissertation. First and foremost, I would like to offer my profound gratitude towards my advisor Prof. (Dr.) Ananda Shankar Chowdhury, for his invaluable guidance, counsel throughout the course of my research work and meticulous correcting my papers. He was very helpful with key suggestions while solving different problems addressed in the thesis. I was fully allowed to have necessary freedom to exercise thoughtful and scientific approaches to the problem and have free discussions on them. He has not only helped me in bringing the thesis to this shape, and also stretched his helping hands whenever I was in need. I consider myself fortunate to have him as my advisor.

I would also like to express my heartfelt gratitude to Prof. Selhi Sinha Chowdhury, Prof. Amit Konar, Prof. Mrinal K. Naskar, Prof. Subir K. Sarkar, Prof. P. Venkateswaran, Prof. Sudipta Chattopadhyay and Prof. Jaydeb Bhaumik for their insightful suggestions in various aspects of this wonderful journey.

I would also like to thank the team at Institute of Neurosciences, Kolkata (IN-K) led by Dr. Mona Tiwari, senior consultant radiologist, for their immense help in providing biological domain knowledge and in formulating solutions of some of the problems related to the completion of this dissertation.

I would also like to thank Dr. Subhrangsu Aditya for introducing me to IN-K and motivating me for pursuing research in this domain. He has been a constant source of motivation and interest for carrying out research work in this domain.

I want to extend my gratitude to the following wonderful members of Imaging, Vision and Pattern Recognition (IVPR) group, who were helpful at various instants Arindam Sikdar, Rukmini Roy, Abhimanyu Sahu, Jagannath Sethi, Sushanta Sahu, Sevakram Kumbhare and Anirban Dutta Choudhury. I would also like to thank Aritro Santra, who also helped in some of the problems related to this dissertation during his internship. Without their help this work would be much difficult to complete.

Last but not the least, I would like to express my heartfelt regards to my mother, Mrs. Meghamala Dey for her lifelong guidance, support and faith that always motivates me to work harder and better. I am thankful to my beloved life partner Ishita Das who believed in me and supported me during the entire course of research study. Without the support of my wife, I would not have completed my Ph.D. work. I would also like to thank my mother in law, Mrs. Rina Das and my father in law, Mr. Litan Das for providing support and believing in me throughout the course of my research work.

Needless to say, without all the above help and support the writing and production this thesis would not have been possible.

Jadavpur University
Kolkata – 700032

Arijit De

Arijit De
Date: 19/09/2024

Contents

List of Figures	xvi
List of Tables	xxii
List of Algorithms	xxiv
1 Introduction	1
1.1 Prologue	1
1.2 Neurological Disorders	2
1.2.1 Brain Tumors	3
1.2.2 Brain Cancers	4
1.2.3 Alzheimer’s Disease	6
1.2.4 Disorders related to Hippocampus	7
1.3 Image Modality	9
1.3.1 Radiological Data	9
1.3.2 Histopathological Data	13
1.4 Basic Theory and Concepts	14
1.5 Brief Overview	15
1.5.1 Brain Tumor Segmentation	16
1.5.2 Brain Tumor Detection and Classification	18
1.5.3 Genetic Biomarker Detection	19
1.5.4 Alzheimer’s Disease Stage classification	20
1.5.5 Hippocampus Segmentation	21
1.6 Research Gaps	23
1.6.1 Brain Tumor Segmentation	23

1.6.2	Brain Tumor Detection and Classification	24
1.6.3	Genetic Biomarker Detection	25
1.6.4	Alzheimer’s Disease Stage classification	26
1.6.5	Hippocampus Segmentation	27
1.7	Organization of Thesis	28
1.8	Contributions of the Thesis	28
2	Theoretical Foundations	33
2.1	Classical Computer Vision Models	33
2.1.1	Graph Cut	33
2.1.2	Graph Cuts for Solving the Max-Flow Min-Cut Problem	35
2.1.3	Multilabel Graph Cuts	39
2.2	Machine Learning Models	41
2.2.1	Random Forest	42
2.2.2	Artificial Neural Networks	45
2.3	Deep Learning Models	50
2.3.1	Convolutional Layer	50
2.3.2	Fully Connected Layer	52
2.3.3	Deep CNN Architectures	54
2.4	Graph based neural networks	58
2.4.1	Tasks	58
2.4.2	Graph Neural Networks	59
2.4.3	Graph Convolutional Networks	60
3	Brain Tumor Segmentation	63
3.1	Introduction	63
3.2	Related Work	64
3.2.1	Classical techniques	64
3.2.2	Deep learned techniques	64
3.2.3	Combination of classical and deep learned techniques	65
3.3	Contributions	66
3.4	Proposed Method	68

3.4.1	3D Brain Tumor Segmentation using UNet	68
3.4.2	3D Brain Tumor Segmentation using Graph Cut	69
3.4.3	3D Brain Tumor Segmentation using Deep Graph Cut	70
3.5	Experimental Results	74
3.5.1	Data Preparation	74
3.5.2	Performance Metrics	74
3.5.3	Ablation Studies	78
3.5.4	Comparison with State-of-the-art Methods	79
3.5.5	Execution Time	79
3.5.6	Time Complexity Analysis	80
3.6	Compliance with ethical standards	80
3.7	Summary	80
4	Brain Tumor Classification	81
4.1	Introduction	81
4.2	Related Works	82
4.3	Contributions	83
4.4	Proposed Method	83
4.4.1	Radiological phase	84
4.4.2	Histopathological phase	86
4.4.3	Graph Convolutional Network	90
4.5	Experimental Results	92
4.5.1	Dataset	93
4.5.2	Training VoxCNN, DenseNet and GCN	93
4.5.3	Ablation Study	93
4.5.4	Comparisons with State-of-the-Art Approaches	94
4.6	Time Complexity Analysis	95
4.7	Summary	96
5	Detection of brain tumor related bio-markers	97
5.1	Introduction	97
5.2	Related Work	99

5.3	Contributions	101
5.4	Methodology	103
5.4.1	Image Feature Extraction	105
5.4.2	Design of Loss Function	106
5.4.3	Training Phase	113
5.4.4	Testing Phase	114
5.5	Experiments	115
5.5.1	Data Preparation	115
5.5.2	Experimental Settings	118
5.6	Results	119
5.6.1	Ablation Study	119
5.6.2	Comparison of Loss Functions	120
5.6.3	State-of-the-art Comparisons	121
5.7	Time Complexity Analysis	123
5.7.1	Time complexity of CNN and FCNs	123
5.7.2	Complexity of Conditional Probability Loss (\mathcal{L}_P)	124
5.7.3	Complexity of Spectral Graph Loss (\mathcal{L}_G)	125
5.7.4	Overall Time Complexity	126
5.8	Summary	126
6	Alzheimer’s Disease Classification	128
6.1	Introduction	128
6.2	Related Work	131
6.2.1	Machine Learning (ML) methods	132
6.2.2	Deep Learning (DL) Methods	132
6.2.3	Data fusion techniques	133
6.3	Contributions	134
6.4	Basics of Diffusion Tensor Imaging	134
6.4.1	Mean diffusivity	136
6.4.2	Fractional anisotropy	136
6.4.3	Echo Planar Imaging	136

6.4.4	Significance of Fractional Anisotropy and Mean Diffusivity . . .	138
6.5	Proposed Method	138
6.5.1	VoxCNN	140
6.5.2	Random Forest Classifier	140
6.5.3	Modulated Rank Averaging based decision level fusion	143
6.6	Experimental Results	146
6.6.1	Data Preparation	146
6.6.2	Implementation Details	149
6.6.3	Time Complexity Analysis	151
6.6.4	Individual Model Performance	152
6.6.5	Combined Model Performance	152
6.6.6	Ablation Study of the Models	154
6.6.7	Comparison with feature fusion strategies	155
6.6.8	Comparison with other approaches	156
6.7	Summary	157
7	Segmentation of the Hippocampus of the human brain	158
7.1	Introduction	158
7.2	Related Works	159
7.3	Contributions	161
7.3.1	Whole Hippocampus Segmentation	161
7.3.2	Hippocampus Subfields Segmentation	161
7.4	Proposed Method for Whole Hippocampus Segmentation	162
7.4.1	Attention UNet	164
7.4.2	Basics of 2D HoG	165
7.4.3	Basics of 3D HoG	165
7.4.4	Majority Pooling	166
7.4.5	A Shape based Loss Function	168
7.5	Hippocampus Subfield Segmentation	169
7.5.1	Deep Graph Cut	169
7.5.2	Multi-class Deep Graph Cut	170

7.5.3	Alpha Beta Swap	171
7.5.4	Deep learned Alpha Beta Swap	172
7.5.5	Deep Learned Shape Information	174
7.5.6	Need of an Adaptive Shape term	174
7.5.7	Shape driven Multi-class Deep Graph Cut	176
7.5.8	Time-complexity Analysis	176
7.6	Data preparation	177
7.6.1	Data Preparation	177
7.7	Implementation Details	179
7.8	Experimental Results	180
7.8.1	Whole Hippocampus Segmentation	180
7.8.2	Hippocampus Subfield Segmentation	181
7.9	Appendix	186
7.10	Summary	187
8	Conclusion	188
8.1	Concluding Remarks	188
8.2	Graphical User Interface	190
8.3	Future Directions	190

List of Figures

1.1	MRI image of an axial slice of brain showing the tumor mass as a bright irregular structure	3
1.2	Different types of glioma tumors as seen from a histopathological image	4
1.3	Difference between a healthy brain and a brain having Alzheimer’s Disease. [1]	6
1.4	Hippocampus inside the brain [2]	8
1.5	Axial FA map of a DTI scan	10
1.6	Axial view of T1, T1-contrast enhanced, T2 and T2-Flair scans. [3] .	13
1.7	Whole slide image of a tumor tissue. H&E stain. The whole slide image (top) is substantially reduced in size digitally from the original scan, but a sample (bottom) is shown at near original resolution. <i>Image courtesy:</i> https://picryl.com/media/whole-slide-image-of-wilms-tumor-974458	14
2.1	Graph construction for a 3×3 image. The dotted line denotes a cut. [4]	34
2.2	An example of segmentation a 3×3 image with graph cut. [5]	37
2.3	Segmentation of the people and bell in the foreground from the background area using graph cut with seeding [6]	38
2.4	Depiction of how the prediction of individual decision tree is used in predicting the final label in a random forest	43
2.5	Biological Neuron Vs Artificial Neuron [7]	47
2.6	Different Neural Activation functions	47
2.7	A typical illustration of a MLP	48
2.8	Illustration of back-propagation through a neuron.	48

2.9	Illustration of Convolutional Layer	49
2.10	Architecture of Alexnet	54
2.11	Architecture of DenseNet 121. Dx: Dense Block x. Tx: Transition Block x.	55
2.12	Architecture of Unet	56
2.13	A single layer of a simple GNN.	60
2.14	Architecture of a Multi-layer Graph Convolutional Network (GCN) with ReLU activation function	62
3.1	(a) A 2D axial slice view of an MRI brain scan, (b) segmented tu- mor region of (a), (c) 2D axial slice of another brain scan, (d) the corresponding segmented tumor output	67
3.2	3D view of the segmented tumor. (a) shows the input 3D MRI volume along with the axial, sagittal and coronal slice views. (b) shows the 3D segmented tumor along with the input brain scan (c) shows the 3D segmented tumor in isolation.	67
3.3	Architecture of 3D UNet. The numbers in each block represent the feature size	69
3.4	Qualitative comparisons for one dataset: (a) Ground Truth (GT), (b) Segmentation using 3D UNet (DSC = 0.862), (c) Segmentation using Graph Cut (DSC = 0.812), (d) Segmentation using DGC (DSC = 0.91)	75
3.5	Qualitative comparisons for a second dataset: (a) Ground Truth (GT), (b) Segmentation using 3D UNet (DSC = 0.877), (c) Segmen- tation using Graph Cut (DSC = 0.897), (d) Segmentation using DGC (DSC = 0.934).	76
4.1	Flowchart of our classification model: 'G' stands for Glioblastoma, 'N' stands for Normal, 'A' stands for Astrocytoma and 'O' stands for Oligodendroglioma.	84
4.2	Overall workflow for radiological classification phase	86

4.3	Axial flair MRI images of (a) Glioblastoma Multiforme, (b) Astrocytoma, (c) Oligodendroglioma. It can be seen in case of (a) the tumor region has thick demarcations with a central dark necrotic region and an irregular whitish edema region while in cases of (b) and (c) there is no such distinctive feature	87
4.4	Histopathology images of (a) normal, (b) glioblastoma, (c) astrocytoma and (d) oligodendroglioma cells with global magnification level. Corresponding images (e), (f), (g) & (h) are shown with local magnification level.	88
4.5	A coarse schematic of a two layer GCN with a fully connected neural network classifier at the end. Source: [8]	92
5.1	Three glioma tumor tissue samples from TCIA [9]. (a) represents a glioblastoma tumor with IDH-wildtype, No 1p/19q codeletion, ATRX-wildtype, TERT mutated and MGMT methylated . (b) represents an Oligodendroglioma tumor with IDH-mutant, No 1p/19q codeletion, ATRX-mutant, TERT unmutated and MGMT methylated. (c) represents an Astrocytoma tumor with IDH-mutant, 1p/19q codeletion, ATRX-wildtype, TERT mutated and MGMT methylated.	99
5.2	Patch generation process: A whole slide image is divided into equal sized square patches. Each patch is scanned and the uninformative patches (shown in gray) are discarded. Remaining patches containing actual tissue images are saved.	102
5.3	Training workflow for a single WSI image at local magnification level. This is repeated for other WSI images in the training set at local magnification level. The whole training process is then repeated at global magnification level.	103

5.4	Overall test time workflow of our model. (a) shows how we get a single probability vector for a single WSI image for local magnification level. This process is repeated at the global magnification level also. (b) shows how we derive the final binary prediction from combining the results of (a).	104
5.5	Graphs in left, middle, and right are for the three glioma tumor tissue samples shown in Fig. 5.1. Here, the vertices are the status of bio-markers, an edge with weight 1 connects two bio-marker vertices having same value as denoted in Table 5.1. An edge with weight e^{-2} means the bio-markers connected by that edge have different values.	110
5.6	The AUC for each bio-marker achieved by the models in shown in Table 5.4. The gray bands represent 95% confidence interval for the AUC of our proposed model. The mean ROC of ML-WCE, ML-WCE + Spectral Loss, ML-WCE + Conditional Probability Loss and the mean ROC of all combined is shown with green, red, blue and magenta lines respectively.	117
6.1	Estimates and probabilistic projections of population of people aged 65 years and above in the world	130
6.2	Axial FA map of a DTI scan from the ADNI dataset.	135
6.3	Axial slices of two patients. (a), (b) (c) denote the EPI, FA and MD maps slices of AD patient respectively and (e), (f), (g) denote that of healthy person. EPI and MD shows the WM in black, while FA shows the WM in white. We can see that in AD patient the total WM region is lesser than normal person which is clearly indicated by the larger size of lateral ventricle body in AD patient	137

6.4	Solution pipeline architecture: C_e, C_{fa}, C_{md} and C_{rfc} denote the probability vectors of EPI, FA, MD and RFC models respectively, each of which contains the probabilities of the four AD classes; Finally, P_i s denote the probabilities for the four AD classes after applying modulated rank averaging. $max(C)$ gives the highest value among the P_i s which corresponds to the AD classes. For example, if the output of $max(C)$ is P_2 , then the AD class corresponding to label 2 is the classification output.	139
6.5	Data set division strategy	148
6.6	Graph showing how number of estimators in Random Forest Classification affects accuracy	150
6.7	Graph showing how number of SMOTE neighbors affects accuracy	151
6.8	Graph showing how number of SURF features affects accuracy	151
6.9	ROC curves of the combined models (Majority Voting, Rank Averaging and Modulated Rank Averaging)	153
7.1	Block diagram of our proposed approach: $pred$ denotes the predicted output of Attention Unet model and gt denotes the ground truth. P_{pred} and P_{gt} are the pooled gradients of predicted and ground truth respectively, as shown in Eq. 7.9.	163
7.2	Architecture of 3D Attention UNet. The input, output and feature dimensions are written beside the corresponding blocks. The number of channels is written on top of each block.	163
7.3	Figure showing the range of angles considered for selecting the voxels from v_θ^s and v_ϕ^s . If α is the most frequently occurring angle, then all angles in range $\alpha \pm \beta$ are considered.	167
7.4	A sagittal slice view of a brain MRI showing the hippocampus bounded in red. The region marked in yellow shows that the contrast is less between hippocampus and its surrounding region, which poses a challenge in segmentation.	174

7.5	Visual comparisons of ground truth segmentation with a model having only dice loss and the same model having the proposed loss function: (a), (b) and (c) represent the axial, sagittal and coronal views of the ground truth HC segmentation, (d), (e) and (f) represent the same views of the predicted segmentation output of the model with dice loss only, (dice score: 0.848), (g), (h) and (i) represent the same views of the predicted segmentation of our proposed loss function (dice score: 0.927). The difference in prediction can be seen in (e) vs. (h) and in (f) vs. (i) and is highlighted with yellow squares and yellow arrows respectively.	182
7.6	Qualitative ablation of our method. GT represents the ground truth. Segmentation with color red represents CA1-3 class, blue represents the CA4/DG class and green represents the SUB class.	184
8.1	Screenshots of the GUI showing the steps of uploading a brain scan, segmentation and then visualizing the segmented tumor	191

List of Tables

3.1	Ablation study I: Graph cut, Graph Cut enhanced with combinations of (a) DGC data term (b) DGC smoothness term (c) DGC variable weight factor. Best values are shown in bold .	78
3.2	Ablation study II: Graph Cut, 3D UNet, DL-GC in 2D and DGC. Best values are shown in bold .	78
3.3	Comparison of State of the art Methods. Best values are shown in bold .	79
4.1	Ablation study of various combinations of modality	94
4.2	Comparison with state of the art models	95
5.1	Details of the genetic markers: ID, Name, and, Interpretation of the values of 1 and 0 following [10].	106
5.2	Table showing the counts of each type of genetic marker. The values and ID are similar to those in the Table. 5.1	116
5.3	Layer architecture of FCN. Each dropout layer drops 10% of the neurons randomly.	118
5.4	Ablation Study: Impacts of individual (ML-WCE) loss, pairwise (Conditional Probability based) loss, and groupwise (Spectral Graph based) loss	120
5.5	Comparison of various loss functions for prediction of five genetic markers	120
5.6	Comparison with state-of-the-art. AUC, Sensitivity (Sens.) and Specificity (Spec.) values are given, as per availability. Best values are highlighted in bold.	121

6.1	Individual evaluation accuracy of EPI, FA, MD and RFC models . . .	153
6.2	Differences in evaluation accuracy at 95% confidence level	154
6.3	Evaluation accuracy and Area under Curve (AUC) of the fusion methods with all four model combinations	155
6.4	Per class metrics containing the precision, recall and f1-score for each disease class	155
6.5	Accuracy and Area Under Curve (AUC) for all combinations using Modulated Rank Averaging method	156
6.6	Comparison of evaluation accuracy with other state-of-the-art approaches	156
6.7	Comparison of evaluation accuracy among different feature fusion methods along with the accuracy (in %) for five types of classifications (AD vs. CN, AD vs. MCI, MCI vs. CN, AD vs. MCI vs. CN, AD vs. EMCI vs. LMCI vs. CN)	157
7.1	Comparison with state-of-the-art methods	180
7.2	Ablation study for the proposed loss function	180
7.3	Ablation Study I: Comparison of segmentation performance of multi-class graph cut, multi-class UNet, and the proposed method. Mean Dice Score of each competing approach over all three classes are reported. Best values are shown in bold	181
7.4	Ablation Study II: Impact of deep learned α - β swap on the segmentation performance. Mean Dice Score of each competing approach over all three classes are reported. Best values are shown in bold	182
7.5	Ablation Study III: Impact of deep learned shape on segmentation performance. Mean Dice Score of each competing approach over all three classes are reported. Best values are shown in bold	183
7.6	Comparison with state-of-the-art methods. Mean Dice Score \pm standard deviation of Dice Score is reported for each class. Additionally, the overall mean Dice Score is reported for each method. Best values are shown in bold	185

List of Algorithms

2.1	Graph Cut Algorithm	37
2.2	α - expansion algorithm	40
2.3	α - β swap algorithm	41
2.4	Random Forest Algorithm	44
3.1	Deep Graph Cut (DGC)	73
5.1	Training Process	113
5.2	Testing Process	114
6.1	MRA(<i>Accuracy</i>)	145
6.2	Algorithm of proposed solution	147
7.1	SMDGC	177

Chapter 1

Introduction

This chapter provides an outline of the importance of neurological disorder detection in the medical imaging community. In section 1.1, we discuss the motivation behind this research work. Section 1.2 presents an overview of neurodegenerative diseases and brain tumors. In section 1.3 we discuss the imaging modalities that we used in our work. Section 1.4 gives an introductory overview about the basic theory and concepts used in this work. We then present a brief overview and background about the works we have done in section 1.5. We examine the research gaps in section 1.6. This is followed by the organization of this thesis in section 1.7. We finally end this chapter, highlighting our contributions in section 1.8

1.1 Prologue

Advances in neuroimaging and machine learning are transforming the landscape of medical diagnostics, particularly in the realm of neurological disorders and brain cancers. This thesis embarks on an ambitious journey to harness the power of state-of-the-art deep learning techniques and innovative algorithmic approaches to enhance the diagnosis and treatment of complex neurological disorders. From the nuanced classification of Alzheimer's Disease stages using sophisticated Diffusion Tensor Imaging scans to pioneering loss functions for precise hippocampus segmentation, this work delves into the intricate interplay between advanced imaging technologies and cutting-edge computational methods.

Furthermore, the thesis introduces novel methodologies for 3D brain tumor segmentation by fusing graph cut and deep learning techniques, and explores multi-modal classification models that significantly elevate the accuracy of glioma diagnosis. By integrating genetic data with histopathological analysis, it also paves the way for predictive models that offer deeper insights into glioma biomarkers, aiming to tailor treatment strategies more effectively.

Through comprehensive experimentation and validation on public datasets, the research presented here not only pushes the boundaries of current diagnostic capabilities but also sets the stage for future innovations in computer-aided diagnosis in neurological disorders. This thesis is a testament to the potential of interdisciplinary approaches in tackling some of the most challenging problems in medical science, promising to inspire further advancements and applications in the field.

1.2 Neurological Disorders

Neurological disorders encompass a wide range of conditions that affect the brain, spinal cord, and nerves. These disorders can have significant impacts on cognitive, motor, and sensory functions. This chapter focuses on brain tumors and brain cancers, followed by a discussion on neurodegenerative diseases like Alzheimer's Disease and brain structures affected by it. Understanding the pathology, symptoms, and current treatment options for these conditions are crucial for advancing medical research and improving patient care. We first discuss brain tumors and brain cancers and how computer-aided intervention and improve the diagnosis and treatment process. Brain tumors and brain cancers are critical conditions that arise from abnormal growths within the brain. They can be benign or malignant, with varying degrees of severity and treatment challenges. Understanding their pathology, symptoms, and management is vital for improving patient outcomes and advancing therapeutic strategies. The following sections discuss brain tumors and brain cancers in detail.

1.2.1 Brain Tumors

Brain tumors are abnormal growths of cells within the brain or central spinal canal, as shown in Fig. 1.1. They can be benign (non-cancerous) or malignant (cancerous). Approximately 700,000 people in the United States are living with primary brain tumors, with about 85,000 new cases diagnosed each year [11]. Brain tumors can originate from brain cells, nerve cells, meninges, or metastasize from other parts of the body. The exact cause of brain tumors is often unknown, but genetic and environmental factors may play a role. Symptoms depend on the tumor's size, type, and location. Common signs include headaches, seizures, cognitive or personality changes, and motor or sensory deficits. Diagnosis is made using neuroimaging techniques such as MRI and CT scans, along with biopsy procedures to determine the tumor's histology. Treatment can be complicated by the tumor's location and its potential impact on critical brain functions. Recurrence is common, and complete resection may not always be possible. Computer vision can assist in the precise delineation of tumor boundaries in neuroimaging, aiding in surgical planning and radiation therapy. Machine learning algorithms can analyze imaging data to differentiate between tumor types and predict growth patterns, improving treatment planning and monitoring.

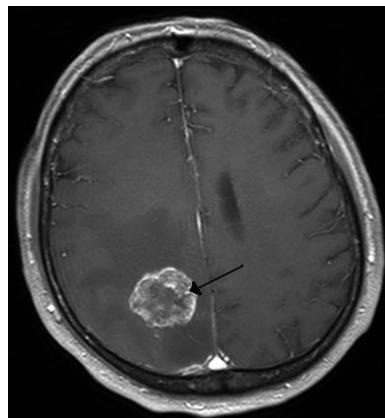
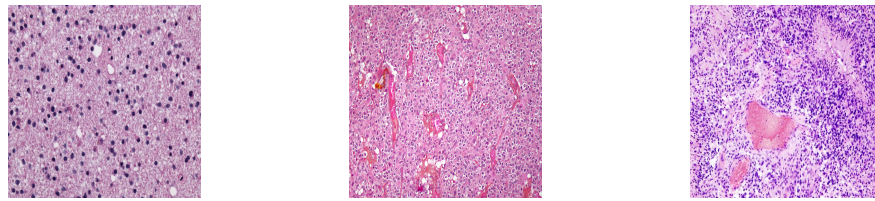


Figure 1.1: MRI image of an axial slice of brain showing the tumor mass as a bright irregular structure



(a) Cellular image of astrocytoma

(b) Cellular image of oligodendroglioma

(c) Cellular image of glioblastoma multiforme

Figure 1.2: Different types of glioma tumors as seen from a histopathological image

1.2.2 Brain Cancers

Brain cancers are malignant tumors that originate in the brain (primary brain cancers) or spread to the brain from other body parts (metastatic brain cancers). Among primary brain cancers, gliomas are the most common and aggressive type, necessitating significant focus and research due to their complex nature and poor prognosis. Gliomas are a diverse group of tumors originating from glial cells in the brain. About 30% of all brain and central nervous system tumors are gliomas. Furthermore, 80% of all malignant brain tumors are also gliomas [12]. The most notable subtypes include:

- **Astrocytomas:** Originating from a type of glial cells in brain called astrocytes, these tumors range from low-grade (pilocytic astrocytoma) to high-grade (glioblastoma multiforme).
- **Oligodendrogliomas:** Derived from oligodendrocytes (a type of glial cells in brain), these tumors often have better prognoses than astrocytomas.
- **Glioblastomas:** The most aggressive subtype, characterized by rapid growth and resistance to traditional therapies, with a median overall survival of approximately 15 months [13].

Histopathology images of the above subtypes are shown in Fig. 1.2, where we can see the cellular structures of the different glioma subtypes. Focusing on gliomas, particularly glioblastomas, is critical due to their aggressive nature, high mortality rates, and the complexity of their treatment. Despite advances in surgical techniques, radiation therapy, and chemotherapy, the prognosis for glioblastoma remains poor,

with approximately 15 months of survival time post-diagnosis. Gliomas represent about 44.6% of all brain tumors in children and adolescents, emphasizing the need for focused research and improved therapeutic strategies [12].

Recent research highlights the importance of genetic biomarkers and mutations in the diagnosis and treatment of gliomas:

- IDH1/IDH2 Mutations: Common in lower-grade gliomas, these mutations are associated with better prognosis and responsiveness to certain therapies [14].
- MGMT Promoter Methylation: The methylation status of the MGMT gene promoter can predict the response to alkylating agents like temozolomide [15].
- 1p/19q Co-deletion: This chromosomal deletion is characteristic of oligodendrogliomas and is associated with a favorable response to therapy and improved survival rates [16].
- TERT Promoter Mutations: These mutations are often found in glioblastomas and are associated with poorer prognosis [17].

Understanding these genetic alterations allows for the development of targeted therapies and personalized treatment plans, improving outcomes for patients with gliomas.

Computer vision techniques play a pivotal role in the detection, diagnosis, and treatment planning of gliomas. Machine learning algorithms can analyze MRI and CT scans to accurately segment tumors, differentiate between tumor types, and assess tumor progression. This technique involves extracting large amounts of quantitative features from medical images, which can be used to predict tumor behavior, treatment response, and patient outcomes. Deep learning models can integrate imaging data with genetic and clinical information to predict prognosis and guide treatment decisions. Augmented reality and computer-assisted navigation systems can help neurosurgeons precisely locate and resect tumors while preserving critical brain functions [18]. The integration of genetic insights and computer vision techniques holds significant promise in enhancing the diagnosis, treatment, and management of glioma tumors. Ongoing research and interdisciplinary collaboration are essential to further improve outcomes for patients with these challenging brain cancers.

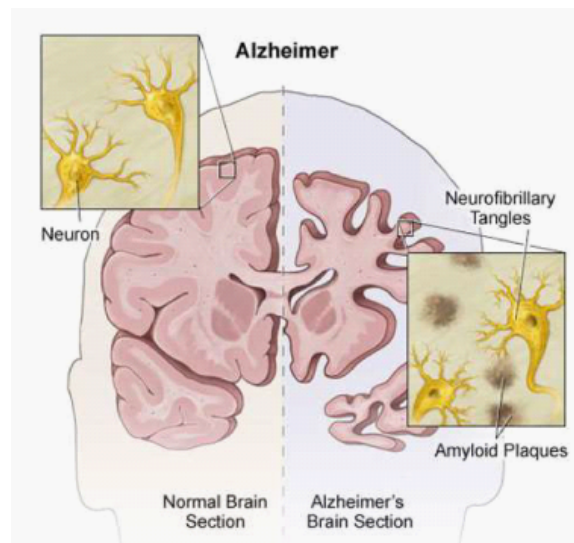


Figure 1.3: Difference between a healthy brain and a brain having Alzheimer's Disease. [1]

We now shift our focus to the Alzheimer's Disease, a prominent neurodegenerative disease. In this context, we also discuss Hippocampus, a structure in the brain, which is often found to be affected by the Alzheimer's Disease, as well as other diseases, such as, Epilepsy.

1.2.3 Alzheimer's Disease

Alzheimer's Disease (AD) accounts for 60-80% of dementia cases worldwide. Dementia is a broad term used to describe a decline in the cognitive function severe enough to interfere with daily life. Alzheimer's Disease leads to progressive degeneration of brain cells, significantly impacting cognitive abilities and memory [19].

Approximately 55 million people worldwide have Alzheimer's disease, a number expected to double every 20 years. By 2050, it is projected that 139 million people will be living with Alzheimer's [20]. The global cost of dementia, largely driven by Alzheimer's, is estimated to be over \$1.3 trillion annually, projected to rise to \$2.8 trillion by 2030 [21].

Alzheimer's disease is marked by the accumulation of amyloid-beta plaques and tau tangles in the brain, as shown in Fig. 1.3. These deposits disrupt cell communication and lead to cell death. The hippocampus, responsible for memory formation,

is one of the first regions to be affected. The cortex, involved in thinking and decision-making, also deteriorates as the disease progresses.

Early symptoms of Alzheimer's include difficulty remembering recent events and names. As the disease progresses, symptoms worsen to include severe memory loss, disorientation, and significant changes in behavior and personality.

Diagnosis of Alzheimer's involves clinical assessments, neuroimaging (such as MRI and PET scans), and biomarker analysis in cerebrospinal fluid. Early and accurate diagnosis is challenging but crucial for managing the disease.

Treatment options for Alzheimer's are limited and primarily focus on slowing disease progression rather than reversing or curing the condition. Current therapies have variable efficacy, and there is a critical need for more effective treatments.

Computer vision can significantly aid in the detection and monitoring of Alzheimer's disease [22]. Advanced image processing techniques can quantify amyloid and tau load from PET scans, providing a more objective measure of disease progression. Automated volumetric analysis of MRI scans can track atrophy in specific brain regions, aiding in early diagnosis and treatment monitoring. These technologies enhance early detection, potentially leading to better management and outcomes for patients.

In summary, Alzheimer's Disease, as the leading cause of dementia, poses significant challenges in terms of diagnosis, treatment, and management. Advances in computer vision and neuroimaging techniques offer promising avenues for improving early detection and monitoring of this debilitating condition.

1.2.4 Disorders related to Hippocampus

The hippocampus is a crucial structure located within the medial temporal lobe of the brain as shown in Fig. 1.4. It plays a significant role in the formation of new memories, spatial navigation, and the consolidation of information from short-term memory to long-term memory. Damage or dysfunction in the hippocampus is associated with several neurological and psychiatric conditions. Diseases affecting the hippocampus include Alzheimer's Disease, epilepsy, depression, and schizophrenia.

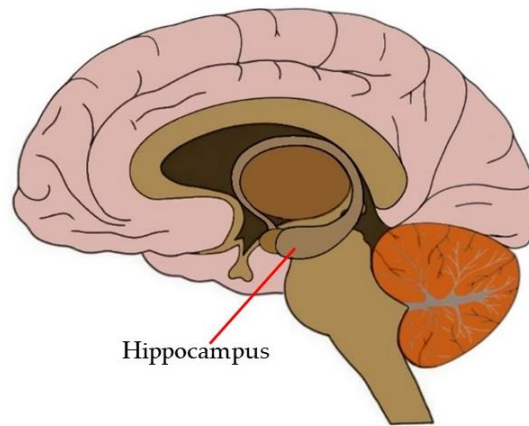


Figure 1.4: Hippocampus inside the brain [2]

Each of these conditions can lead to significant cognitive and behavioral impairments, highlighting the importance of the hippocampus in brain function [23].

Alzheimer’s Disease (AD) is the most well-known condition associated with hippocampal damage. AD is characterized by the accumulation of amyloid-beta plaques and tau tangles, leading to neurodegeneration in the hippocampus and other brain regions. This degeneration results in progressive memory loss, disorientation, and cognitive decline [24].

Over 50 million people worldwide are affected by epilepsy, making it one of the most common neurological diseases globally [25]. Epilepsy, particularly temporal lobe epilepsy (TLE), often involves the hippocampus. TLE is characterized by recurrent seizures originating in the temporal lobe, where the hippocampus is located. These seizures can cause hippocampal sclerosis, a condition marked by neuronal loss and gliosis (scarring) in the hippocampus. Patients with TLE often experience memory deficits and spatial disorientation [26].

The hippocampus is highly susceptible to damage due to its unique structure and function. In Alzheimer’s Disease, the accumulation of amyloid-beta plaques and tau tangles disrupts neuronal communication and leads to cell death, particularly affecting the hippocampus [24]. In epilepsy, recurrent seizures cause excitotoxicity, leading to neuronal loss and gliosis in the hippocampus [26].

The symptoms associated with hippocampal damage vary depending on the specific disease but often include memory loss, disorientation, and cognitive impair-

ments. In Alzheimer’s Disease, patients experience progressive memory decline and difficulty with spatial navigation. In epilepsy, patients may have memory deficits and confusion following seizures.

Diagnosing hippocampal-related diseases involves a combination of clinical assessments, neuroimaging (such as MRI and PET scans), and neuropsychological testing. Biomarker analysis in cerebrospinal fluid can also aid in diagnosing conditions like Alzheimer’s Disease.

Despite available treatments, many patients continue to experience seizures. Identifying the exact brain regions responsible for seizures can be difficult, making targeted treatments challenging. Computer vision techniques can enhance the segmentation of hippocampus and analyse the degree of damage by comparing it to a healthy hippocampus through detailed analysis of neuroimaging data. Automated analysis of MRI data can improve the accuracy of diagnosis and the identification of seizure onset zones, potentially leading to more effective surgical interventions.

1.3 Image Modality

This section provides a comprehensive overview of the image modalities used in the thesis, focusing on radiological and histopathological data. The radiological data includes Diffusion Tensor Imaging and various Magnetic Resonance Imaging techniques, while the histopathological data consists of H&E stained Whole Slide Images of brain tissue. Each modality’s principles, features, capture methods, and common file formats are discussed.

1.3.1 Radiological Data

(A) Diffusion Tensor Imaging

Diffusion Tensor Imaging (DTI) is an advanced MRI technique that measures the diffusion of water molecules in biological tissues. It is particularly effective for visualizing the orientation and integrity of white matter tracts in the brain. DTI is captured using MRI scanners equipped with diffusion-sensitized gradients. These gradients measure the diffusion of water in multiple directions, producing data that

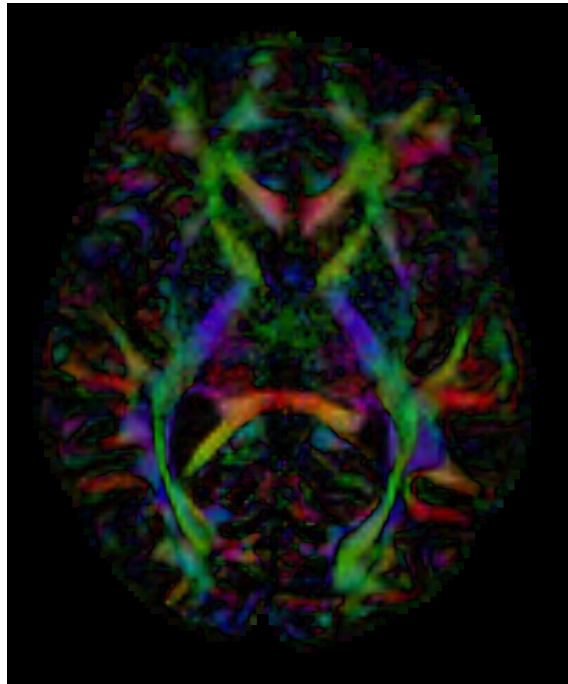


Figure 1.5: Axial FA map of a DTI scan

is processed to generate diffusion tensors, which describe the magnitude and directionality of diffusion [27]. DTI has the following features that are used in our work.

- **Fractional Anisotropy:** Measures the degree of anisotropy of water diffusion, indicating the integrity of white matter tracts.
- **Mean Diffusivity:** Represents the average rate of water diffusion within tissue, useful for identifying tissue abnormalities.
- **Echo Planar Imaging:** It is an MRI sequence that allows for the rapid acquisition of images by capturing all spatial frequencies required for image reconstruction following a single excitation pulse. The primary advantage of EPI is its speed, making it suitable for applications where rapid imaging is essential, such as DTI.

DTI data is typically stored in formats such as NIfTI (.nii, .nii.gz) or DICOM, which support multi-dimensional data necessary for 3D imaging. An example DTI image is shown in Fig. 1.5.

(B) Magnetic Resonance Imaging (MRI)

MRI includes several techniques, each providing different tissue contrasts and information.

T1 weighted Imaging T1-weighted images are produced by exploiting the differences in longitudinal relaxation times (T1) of tissues. This is achieved by applying a radio frequency pulse and measuring the time it takes for protons to realign with the magnetic field. T1-weighted images are captured using spin-echo sequences or gradient-echo sequences on MRI scanners. T1 weighted images have the following key features-

- **High-resolution anatomical details:** Excellent for visualizing the structure of the brain.
- **Contrast between different tissue types:** Useful for identifying normal and pathological structures.

T1-weighted images are commonly stored in DICOM format for clinical use and NIfTI format for research purposes. T1-weighted MRI is crucial in detecting brain parenchymal and meningeal abnormalities, enhancing the diagnostic accuracy in clinical practice [28].

T2-weighted Imaging Principles and Capture Method: T2-weighted images highlight differences in transverse relaxation times (T2), where tissues with high water content appear bright. These images are captured using spin-echo or fast spin-echo sequences on MRI scanners. The important features of T2 imaging are specified below-

- **High sensitivity to fluid content:** Effective for detecting edema, inflammation, and other fluid-related abnormalities.
- **Clear differentiation between gray and white matter:** Useful for identifying lesions and structural abnormalities.

T2-weighted images are also typically stored in DICOM and NIfTI formats. T2-weighted MRI has been shown to be effective in differentiating high-grade gliomas from metastases, providing valuable diagnostic information [29].

T1 Contrast-Enhanced Imaging This technique involves the use of gadolinium-based contrast agents to enhance the visibility of certain structures or abnormalities by shortening the T1 relaxation time of tissues. Images are captured using spin-echo or gradient-echo sequences post-contrast administration. The noteworthy characteristics of T1 Contrast Enhanced Imaging are given below-

- **Enhanced contrast between normal and pathological tissues:** Improves detection and characterization of tumors, vascular abnormalities, and inflammation.
- **Dynamic imaging capabilities:** Useful for assessing blood-brain barrier integrity and perfusion.

Contrast-enhanced images are stored in DICOM and NIfTI formats. Enhanced T1 imaging has proven superior in identifying abnormalities in brain parenchymal and meningeal diseases [28].

T2 - FLAIR (Fluid-Attenuated Inversion Recovery) Principles and Capture Method: T2-FLAIR imaging suppresses the signal from cerebrospinal fluid, making it easier to detect lesions near fluid-filled spaces. This is achieved by applying an inversion recovery pulse before the standard T2-weighted sequence. The characteristics of T2-FLAIR is mentioned below-

- **Suppression of CSF signal:** Enhances the visibility of lesions in the periventricular region and around the brain's fluid spaces.
- **High sensitivity to white matter lesions:** Useful for diagnosing multiple sclerosis, epilepsy, and other conditions involving white matter abnormalities.

T2-FLAIR images are typically stored in DICOM and NIfTI formats. T2-FLAIR imaging provides significant diagnostic benefits in detecting meningeal lesions and

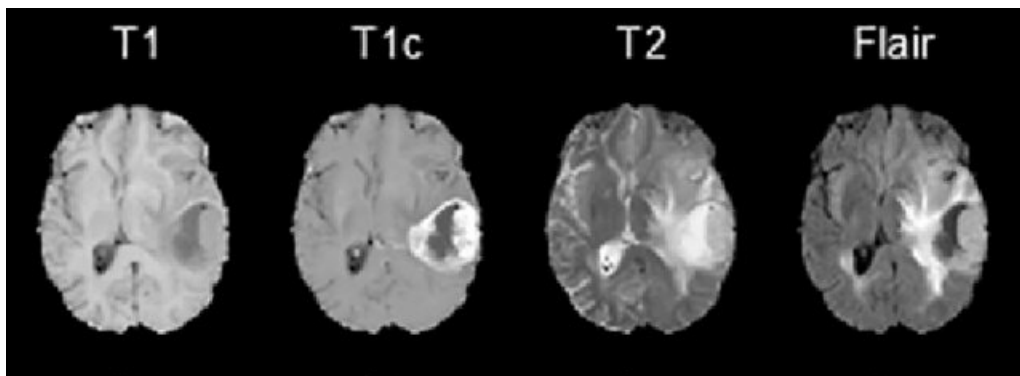


Figure 1.6: Axial view of T1, T1-contrast enhanced, T2 and T2-Flair scans. [3]

differentiating between various brain pathologies [30]. An example slice image of the different types of MRI images discussed above are shown in Fig. 1.6.

1.3.2 Histopathological Data

(A) H&E Stained Whole Slide Images (WSI)

Hematoxylin and Eosin (H&E) staining is a routine technique in histopathology that provides detailed visualization of tissue architecture [31]. Hematoxylin stains cell nuclei blue, while eosin stains the extracellular matrix and cytoplasm pink. WSIs are created by scanning stained glass slides at high resolution using digital pathology scanners, which capture the entire tissue section in a digital format. The key features of this image modality are as follows-

- **Detailed tissue architecture:** Provides a comprehensive view of tissue morphology, essential for diagnosing various diseases.
- **High-resolution imaging:** Allows for detailed examination of cellular and tissue structures. See Fig. 1.7 for reference.

WSIs are commonly stored in formats such as .svs, .ndpi, .tiff, and .vms, which support high-resolution and large file sizes necessary for detailed pathology images.

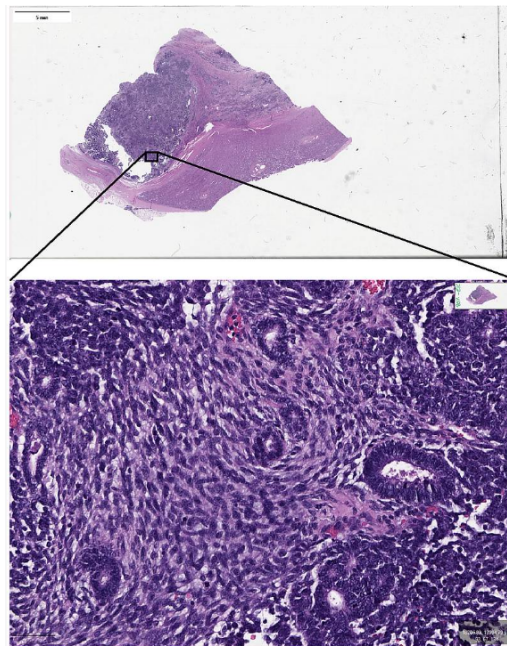


Figure 1.7: Whole slide image of a tumor tissue. H&E stain. The whole slide image (top) is substantially reduced in size digitally from the original scan, but a sample (bottom) is shown at near original resolution. *Image courtesy:* <https://picryl.com/media/whole-slide-image-of-wilms-tumor-974458>

1.4 Basic Theory and Concepts

In this section, the theoretical foundations and key concepts in computer vision, machine learning, and deep learning, integral to the research work carried out in the thesis, are presented.

We first discuss classical computer vision models, focusing on graph cuts. We explore the graph cut method for image segmentation, delving into fundamental concepts like min-cut/max-flow problems, modifying energy functions with data and smoothness terms, and the critical role of seeding in segmentation accuracy. We also discuss extensions to multi-class problems, focusing on algorithms like alpha-expansion and alpha-beta swap for handling multiple labels, and illustrate their applications in various vision tasks. We have used graph cuts in more than one of our works. For brain tumor segmentation, we used graph cuts to complement the drawbacks of deep learning architecture called U-Net. For Hippocampus segmentation, we used multi-class graph cuts to achieve fine grained multi-class segmentation.

Next, we briefly discuss about the machine learning models we used in our work,

particularly the Random Forest algorithm. The ensemble learning approach, where multiple decision trees are built on random data subsets, and predictions are aggregated for improved accuracy, is used in our work on Alzheimer’s Disease stage classification. The Random Forest’s working mechanism, advantages such as handling high-dimensional data and preventing overfitting, and its applications in different industries like healthcare and finance makes it a good classifier for handling multiple feature based data in our work.

We then discuss about deep learning models used in our work. Prominent CNN architectures that we used are:

- **DenseNet:** Known for its dense connectivity, reducing the vanishing gradient problem and improving feature reuse. We used this as a deep feature extractor for histopathological images in our work on brain tumor classification [32].
- **U-Net:** Designed for biomedical image segmentation, notable for its symmetric U-shaped structure and skip connections for precise localization [33]. We used UNet and its variants in multiple works like Brain tumor segmentation, Hippocampus segmentation and multi-class Hippocampus segmentation.

Finally, the thesis introduces graph-based neural networks, particularly Graph Convolutional Networks (GCNs) [34], which is like the extension of traditional neural networks to handle graph-structured data by redefining the convolution operation for nodes and their connections. We have used GCNs in brain tumor classification tasks and was able to achieve state of the art results with it.

The details of each model and algorithm used in the thesis is explained in detail in Chapter 2 with visual illustrations and captivating descriptions along with the mathematical foundations.

1.5 Brief Overview

This section gives a broad overview on the related works of various neurological disorder detection and analysis problems addressed in this thesis.

1.5.1 Brain Tumor Segmentation

Brain tumor segmentation methods can be broadly classified into three groups, namely, Without Learning, Supervised Learning, and Unsupervised Learning methods.

(A) Methods Without Learning

Some examples of this class are - thresholding and region growing techniques.

Thresholding Thresholding is a simple, fast technique that converts a scalar image into a binary image based on a threshold value, separating foreground and background regions. The threshold value is iteratively updated based on the mean intensities of the partitioned regions until it converges [35, 36].

Region Growing Region growing initializes with a seed point and groups neighboring pixels with similar properties to form regions. This method is computationally simple but sensitive to seed initialization and noise [37].

(B) Supervised Learning Methods

Supervised methods use labeled training data to build models that classify pixels during testing. These methods include clustering algorithms and machine learning techniques, like, Support Vector Machines (SVM) and Artificial Neural Networks (ANN) [38].

Artificial Neural Networks (ANN) Artificial Neural Networks (ANNs) mimic the behavior of biological neurons, processing input signals and transmitting them through multiple hidden layers to produce a final output. The output of each artificial neuron is a non-linear function of the sum of its inputs, and the network learns by adjusting the weights of the connections between neurons to minimize errors [39].

Convolutional Neural Networks (CNNs) Convolutional Neural Networks (CNNs) are a type of ANN specifically designed for processing structured grid

data like images. CNNs automatically detect and learn features from input images through different types of layers. Detailed explanation of CNNs can be found in Chapter 2.

Deep Neural Networks (DNNs) Deep Neural Networks (DNNs) extend the concept of ANNs by adding more hidden layers, allowing them to model more complex functions. However, DNNs require a large amount of training data and substantial computational resources, which can be a limitation in medical imaging applications [40,41]. More details on DNN is provided in Chapter 2

(C) Unsupervised Learning Methods

Unsupervised methods do not require labeled training data. They automatically group pixels into clusters based on image features like intensity and texture. Techniques include k-means clustering and Active Contour Models (ACM) [42].

Hybrid Techniques Hybrid techniques combine multiple methods to leverage their respective advantages. Examples include integrating Fuzzy C-means (FCM) with SVM for improved classification accuracy [43].

k-Means Clustering k-means clustering partitions image data into k clusters, iteratively updating cluster means to minimize within-cluster variance. This method is efficient but may not always converge to the optimal solution [44].

Active Contour Models (ACM) ACMs are deformable models that evolve to match object boundaries based on image gradients and external forces. They can handle significant variability but require good initialization and can be computationally intensive [45].

Combining Methods Combining methods like FCM and SVM provides accurate classification and efficient segmentation by balancing sensitivity to noise and preserving image information [46].

1.5.2 Brain Tumor Detection and Classification

Brain tumor classification using MR images is a challenging and essential task in medical image analysis. Various methods have been developed over the years to enhance the accuracy and efficiency of brain tumor classification.

(A) Traditional Machine Learning Methods

Preprocessing, feature extraction, feature selection, dimension reduction, and classification are typical phases in machine learning approaches for classification. Since the feature extraction stage frequently requires specialist knowledge, it might be difficult for non-experts to use these techniques efficiently. Two types of features can be extracted: high-level (local) features like Bag-of-Words (BoW), Fisher Vector (FV), and Scale-Invariant Feature Transform (SIFT) and low-level (global) features like intensity and texture characteristics. [47], [48], [49].

(B) Deep Learning Methods

The advent of deep learning has significantly transformed the landscape of medical image classification. CNNs are particularly successful due to their ability to learn feature representations directly from the data, bypassing the need for manual feature extraction. Deep learning approaches do not require handcrafted features and can automatically generate powerful discriminating features using a hierarchical learning approach. This capability makes CNNs highly effective in capturing both low-level and high-level features from MR images [50–54].

To address the limitations posed by small datasets, transfer learning has emerged as a viable solution. Transfer learning involves using a pre-trained CNN model on a large dataset and fine-tuning it for the specific task at hand. This approach leverages the knowledge learned from natural images and adapts it to medical images, thus improving performance even with limited data. Studies have shown that fine-tuning pre-trained models like VGG19 can significantly enhance classification performance for brain tumor MR images [55–57].

1.5.3 Genetic Biomarker Detection

Biomarker detection is crucial for the diagnosis, prognosis, and treatment of various diseases, especially brain cancer. Recent advances in machine learning (ML) and deep learning (DL) have significantly enhanced the ability to identify relevant biomarkers from complex datasets. Although not much work in this direction with regards to brain tumor, we discuss some works done for other diseases along with the works done for brain tumor bio markers that can help provide the basic background and motivation for this type of research.

We now discuss the works categorising it into three types of bio markers - diagnostic, prognostic and predictive.

(A) Diagnostic Biomarkers

Diagnostic biomarkers are essential for confirming the presence of a disease and distinguishing between different subtypes of cancer. ITIH5 and DKK3 were found to be probable biomarkers with a precision of 93% by Klotten *et al.* pursuing promoter methylation of seven putative tumour suppressor genes in breast cancer [58]. Using LASSO, Chi-Squared, and Information Gain approaches for feature selection and RF and SVM for classification, Rehman *et al.* employed machine learning algorithms to validate the significance of miRNAs as breast cancer biomarkers [59].

(B) Prognostic Biomarkers

Prognostic biomarkers forecast patient outcomes and illness recurrence. Using clinical and RNA-seq data from the TCGA portal, Ma *et al.* developed ML methods to uncover 16 gene prognostic markers for lung adenocarcinoma (LUAD), and they validated their findings with GEO datasets [60]. In order to find prognostic small nucleolar RNAs (snoRNAs), Xing *et al.* employed survival-related Cox regression analysis models [61]. They chose candidates using LASSO regression and validated using multivariate Cox proportional hazard models. Wong *et al.* applied a deep multilayer perceptron network to identify prognostic genes for glioblastoma (GBM), achieving significant results in survival analysis [62].

(C) Predictive Biomarkers

Predictive biomarkers help classify individuals likely to respond to specific treatments. Nam *et al.* developed the Gene Ranker method to identify predictive markers from gene expression data, integrating networks using WGCNA and generating scores for genes such as OTC, B3GNT9, and Clorf167 [63]. Zhao *et al.* introduced a graph convolutional network (GCN) to prioritize protein-coding genes using lncRNAs datasets, achieving high accuracy in AUC and AUPR values [64]. Zhang *et al.* proposed a network-based deep learning approach for gene prioritization, constructing a human molecular interaction network and training it with GNN to identify key genes [65].

Biomarker detection in brain tumors has gained significant attention due to its potential to improve diagnosis and treatment. Yan *et al.* used a deep learning approach to identify glioblastoma biomarkers from MRI data, achieving high accuracy in distinguishing tumor subtypes [66]. Li *et al.* developed a machine learning model that integrates genomic and clinical data to predict patient outcomes in glioma, identifying key biomarkers such as IDH1 and MGMT [67]. Chen *et al.* applied a convolutional neural network (CNN) to RNA-seq data to discover biomarkers for brain metastases, demonstrating the method's potential in clinical settings [68]. Furthermore, recent studies have highlighted the use of liquid biopsies, where circulating tumor DNA (ctDNA) and extracellular vesicles (EVs) are analyzed using ML techniques to identify brain tumor-specific biomarkers [69].

1.5.4 Alzheimer's Disease Stage classification

Recently, there are more reports on detecting Alzheimer's Disease by using MRI data and machine learning (ML) techniques and Deep Learning (DL) Techniques. We first discuss the ML techniques used, followed by DL techniques.

(A) Traditional Machine Learning Techniques

Early studies applying machine learning (ML) to Alzheimer's Disease (AD) focused on traditional algorithms such as Decision Trees (DT), Support Vector Machines

(SVM), Naïve Bayes (NB), and k-nearest neighbors (k-NN). These methods used manually selected linguistic features from radiological data. For instance, Guinn *et al.* compared DT, SVM, and k-NN on conversational samples [70]. Orimaye *et al.* evaluated DT, NB, SVM with radial basis function, and neural networks, concluding that SVM performed best [71]. Yancheva *et al.* applied random forest (RF) classifiers using automatically extracted semantic features [72]. Decision Trees (DT) are a non-parametric supervised learning method used for classification and regression. They create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features [73]. Breiman *et al.* used Random Forest (RF), which is an ensemble learning method for classification and regression that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes or mean prediction of the individual trees [74].

(B) Deep Learning Techniques

Recent advancements in neural networks have significantly enhanced Alzheimer’s Disease (AD) classification. Orimaye *et al.* used a deep-deep neural network language model (D2NNLM) with high-order n-gram features to distinguish between AD and control samples [75]. Karlekar *et al.* applied convolutional neural networks (CNN), long short-term memory (LSTM) networks, and hybrid CNN-LSTM models for the same task. These results indicate that neural networks outperform traditional ML algorithms in AD classification due to their ability to automatically learn complex patterns from large datasets.

1.5.5 Hippocampus Segmentation

Hippocampus segmentation has been a critical area of research due to its implications in diagnosing and understanding various brain diseases, such as Alzheimer’s disease, temporal lobe epilepsy, and schizophrenia. Traditionally, segmentation methods have been categorized into manual, semiautomatic, and fully-automated approaches. Manual segmentation is the gold standard, but is labor-intensive and

time-consuming. Semiautomatic methods require user interaction, whereas fully-automated methods aim to eliminate this need. We discuss below four different categories of segmentation approaches.

(A) Atlas-Based Segmentation

Atlas-based segmentation methods are quite popular. By lining up the target image and the labelled atlas image, Fischl *et al.* [76] were able to segment different brain areas. Multi-atlas-based techniques have been developed as a response to the shortcomings of single atlas registration. These approaches entail registering each atlas picture to the target image and then combining all propagating atlas labels to produce the final segmentation result. Researchers have concentrated on enhancing label fusion approaches and image registration procedures to improve multi-atlas segmentation performance [77].

(B) Classification-Based Segmentation

Another approach is classification-based segmentation, where the segmentation task is transformed into a pattern recognition problem. This method involves feature extraction from the target image followed by the use of supervised learning algorithms such as random forests and support vector machines (SVM) to build a classification model. Researchers have introduced techniques like multi-scale and sparse representation to improve feature representation and classifier performance [78] [79].

(C) Deep Learning-Based Segmentation

Deep learning techniques have recently demonstrated better results in tasks involving the segmentation of medical images. Particularly effective are Convolutional Neural Networks (CNNs) and U-net designs. Hippocampal segmentation research has been resurrected by U-Net [33], which has greatly increased segmentation speed and precision. Because deep neural networks can automatically learn complicated patterns from vast datasets, studies have shown that they perform better than traditional machine learning techniques [80].

(D) Hybrid Approaches

A combination of classical and deep learning (DL) techniques can yield superior segmentation performance compared to using DL or classical methods alone. Published works, such as, [81] and [82] demonstrate this for lung nodule segmentation. In case of 3D brain tumor segmentation [83], integrating UNet with graph cut addressed the manual seeding issue of graph cut and the undersegmentation problem of UNet caused by limited data.

1.6 Research Gaps

This section highlights the existing research gaps in the specific problem domains addressed in this thesis. These gaps reveal the importance of the problems we aim to solve.

1.6.1 Brain Tumor Segmentation

Despite the advancements in brain tumor segmentation methods, several research gaps persist that limit the efficacy and generalization of these techniques in clinical applications.

Firstly, classical methods like thresholding and region growing, while tend to be computationally efficient and simple, suffer from significant limitations. These methods are highly sensitive to initial conditions, such as, seed selection in region growing and threshold values in thresholding. They also struggle with segmenting tumors that have irregular shapes, varying intensities, or that are located near complex anatomical structures. The reliance on manually defined parameters makes these methods less adaptable to the diverse range of tumors encountered in clinical practice. Therefore, there is a need for more adaptive and robust methods that can automatically adjust to different tumor characteristics.

Secondly, while supervised learning methods, particularly machine learning techniques like SVM and ANN, have shown promise in improving segmentation accuracy, they rely heavily on the availability of large, well-annotated datasets for training.

The scarcity of such datasets in medical imaging, combined with the time-consuming and subjective nature of manual annotation, poses a significant challenge. Hybrid techniques that combine the strengths of both classical and deep learning approaches show potential for improving segmentation accuracy and robustness. However, the optimal integration of these methods remains an open research question. Some challenges in choosing and combining different methods includes understanding how to balance the trade-offs between computational efficiency, segmentation accuracy, and robustness to variations in tumor presentation.

Finally, a major gap in current research is the ability of segmentation methods to handle particularly challenging cases, such as tumors with highly irregular shapes, heterogeneous textures, or those adjacent to critical brain structures. Most existing algorithms are designed for typical tumor appearances and may fail in these complex scenarios. Therefore, developing more sophisticated algorithms that can effectively segment such difficult cases is crucial.

Addressing these research gaps will be essential for advancing the field of brain tumor segmentation and translating these methods into reliable tools for clinical use, ultimately improving patient outcomes through more accurate diagnosis and treatment planning.

1.6.2 Brain Tumor Detection and Classification

Despite the advancements in brain tumor classification using MR images, several critical research gaps need to be addressed to improve the reliability and applicability of these methods in clinical settings.

Firstly, traditional machine learning methods, while effective, often involve complex preprocessing steps such as feature extraction, feature selection, and dimensionality reduction. These steps require expert knowledge, making them difficult to implement for non-experts in the field. Additionally, the reliance on handcrafted features, whether low-level like intensity and texture or high-level, like BoW and SIFT, can limit the adaptability of these methods to different datasets or imaging conditions. The challenge lies in automating these processes or developing more

generalized approaches that can perform well across a variety of imaging conditions without the need for extensive manual intervention.

Secondly, deep learning methods, particularly CNNs, have shown considerable promise in brain tumor classification due to their ability to learn features directly from data. However, these methods face significant challenges related to overfitting, especially when trained on small datasets. While CNNs excel at capturing both low-level and high-level features, they often require extensive computational resources and time for training, particularly as model architectures become more complex. This can limit the accessibility of these methods, especially in resource-constrained environments.

Another critical research gap is the underutilization of multimodal data in brain tumor classification. Recent research has demonstrated that using multi-modal data can induce complementary features that can be learned by deep learning models to provide better classification accuracy [84]. We have explored the use of multi modal data in our works.

Addressing these research gaps will be crucial for advancing the field of brain tumor classification and ensuring that these methods can be effectively integrated into clinical workflows, ultimately leading to better diagnosis, treatment planning, and patient outcomes.

1.6.3 Genetic Biomarker Detection

Despite the promising advancements in biomarker detection for brain tumors, particularly through the integration of deep learning and machine learning techniques with genetic and genomic data, several key research gaps persist.

Firstly, the identification of genetic mutations as biomarkers for brain tumors, such as IDH1 and MGMT in gliomas, has significantly advanced our understanding of tumor subtypes and patient prognosis. However, most existing studies have primarily focused on a limited set of well-known genetic mutations. There is a need for more comprehensive investigations that explore a broader spectrum of genetic alterations, including rare mutations and epigenetic modifications, which could provide

additional prognostic and therapeutic insights.

Finally, the integration of multi-omics data (e.g., genomic, transcriptomic, and proteomic data) with clinical data presents a powerful approach for biomarker discovery, as demonstrated by Li *et al.* [67] in glioma prognosis. However, there are challenges related to the standardization and harmonization of multi-omics data, particularly when integrating datasets from different platforms or institutions. There is a critical need for the development of robust computational frameworks and data-sharing platforms that can effectively integrate and standardize multi-omics data, enabling more reliable biomarker discovery.

1.6.4 Alzheimer’s Disease Stage classification

Despite the significant advancements in applying machine learning (ML) and deep learning (DL) techniques for Alzheimer’s Disease (AD) stage classification, several critical research gaps persist.

First, while much of the existing research has focused on classifying AD stages, there is a notable lack of studies specifically targeting the early stages of the disease, such as, early mild cognitive impairment (EMCI), and late mild cognitive impairment (LMCI). Identifying these stages is crucial, as early intervention can significantly slow the progression of AD. The difficulty in classifying these early stages stems from the subtlety of symptoms and the overlap with normal aging processes, which requires the development of more sensitive and specific models.

Second, most studies to date have focused on classifying AD stages in isolation, typically distinguishing between binary stages such as AD vs. non-AD or MCI vs. healthy controls. There is a significant gap in research that attempts to classify multiple stages of AD simultaneously. Multi-stage classification is essential for a detailed understanding of disease progression and could improve the precision of interventions tailored to specific stages of AD.

Finally, while CNNs and LSTMs have been combined in hybrid models to capitalize on their respective strengths, the exploration of other hybrid or ensemble techniques, particularly those that might integrate traditional ML methods with

DL approaches, remains sparse. Such models could potentially improve classification accuracy and generalization across different datasets.

Addressing these gaps could lead to significant advancements in the early detection and stage-specific treatment of Alzheimer’s Disease, ultimately improving patient outcomes.

1.6.5 Hippocampus Segmentation

Despite the progress in hippocampus segmentation techniques, several issues remain unaddressed, limiting the potential of these methods in clinical applications.

Firstly, although atlas-based segmentation methods have been extensively studied, their performance is highly dependent on the quality of the image registration and label fusion processes. The challenge arises in accurately aligning the atlas images with target images, especially when there are significant anatomical variations among individuals. While multi-atlas strategies attempt to mitigate this by incorporating multiple references, there is still a need for more robust and flexible registration techniques that can handle diverse and complex brain structures. Additionally, the computational cost and time required for multi-atlas approaches remain a concern, particularly in large-scale studies or real-time clinical settings.

Secondly, although deep learning-based segmentation methods, particularly CNNs and U-Net architectures, have revolutionized the field by automating feature learning and significantly improving segmentation accuracy, they are not without limitations. One major challenge is the need for large, annotated datasets to train these models effectively. Such large datasets are often unavailable in the field of medical imaging because of privacy issues and the labor-intensive nature of human annotation. The fact that deep learning models have a tendency to overfit on limited datasets, which results in poor generalization on new data, exacerbates this problem. Furthermore, while deep learning models excel at capturing complex patterns, they can sometimes struggle with the fine details necessary for precise hippocampus segmentation, particularly in cases of subtle pathological changes.

Another research gap lies in the integration of classical and deep learning tech-

niques. Although preliminary studies suggest that combining these methods can lead to improved segmentation performance, the optimal way to fuse classical approaches with deep learning remains an open question. For instance, while hybrid methods like the integration of U-Net with graph cut techniques have shown promise, they still face challenges such as balancing the strengths of each method and addressing issues like under-segmentation and the need for manual intervention. Addressing these gaps will be critical for advancing the accuracy, efficiency, and clinical utility of hippocampus segmentation methods, ultimately improving diagnosis and treatment planning for various neurological conditions.

1.7 Organization of Thesis

We organize the thesis in the following manner: In **chapter 2** we have stated the fundamental concepts and theoretical background of different models utilized in solving different brain disorder problems. In **chapter 3** we have addressed the issue of 3D brain tumor segmentation using a synergism of classical and deep learned techniques. In **chapter 4** we have proposed a novel brain tumor classification using multi modal data. In **chapter 5**, we designed a “multi-label” brain tumor bio marker prediction framework using histopathology data. In **chapter 6** we have provided a detailed description of DTI based Alzheimer’s disease detection model. In **chapter 7** we have proposed a “shape-driven” hippocampus segmentation framework using radiology data and also extended this model to segment important sub regions inside the hippocampus. In **chapter 8**, we finally conclude this thesis and discuss some of the future directions of current neurological disorder problems with high practical utility.

1.8 Contributions of the Thesis

In this thesis, we have five major contributions. We mostly decide upon the problems in the domain of brain disorders after careful discussion with clinicians at the Institute of Neurosciences, Kolkata by considering factors like real-world challenges faced

during patient analysis, treatment and diagnosis. We now present each contribution in the following paragraphs.

Brain tumor segmentation is crucial for tumor diagnosis and surgical planning. A common challenge encountered by radiologists and neurosurgeons is accurately delineating the boundaries of a tumor when the abnormality is not well-defined and spreads into the surrounding tissue. Neuro-radiologists desire a concise yet precise method for identifying tumors that can expedite the preoperative diagnosis. To tackle the challenge of effective 3D tumor segmentation, we propose a novel approach to 3D brain tumor segmentation using a combination of deep learning and graph cut techniques on MRI data. Specifically, probability maps generated by a UNet model for classifying voxels as tumor or background are utilized to enhance the graph cut energy function. We introduce new expressions for the data term, region term, and weight factor that balance these terms for individual voxels in our model, detailed description of which is available in Chapter 3. The performance of our approach is validated using the BRATS 2018 dataset, demonstrating superior segmentation accuracy compared to using graph cut or UNet independently, as well as other state-of-the-art methods.

Histopathological classification of brain tumors relies on identifying specific histopathological features unique to each tumor type. Accurate classification of gliomas into astrocytoma, oligodendroglioma, and glioblastoma is crucial for patient prognosis, treatment, and management. While microscopic examination is the gold standard, it is subject to significant inter-observer variability. Machine learning algorithms that utilize specific histopathological characteristics can help reduce this variability by providing objective analysis. This work addresses brain tumor classification using both radiology and histopathology data through a coarse-to-fine approach combining deep learning and Graph Convolution Networks (GCNs). Initially, a 3D CNN is employed to detect glioblastoma from MRI images. For distinguishing astrocytoma and oligodendroglioma, Whole Slide Images (WSIs) are used in the subsequent stage. During this fine classification phase, 2D CNN features are extracted at two different magnification levels, constructing a graph with nodes as concatenated

feature embeddings and edges determined by a Radial Basis Function (RBF) kernel. The GCN, utilizing a normalized graph Laplacian, ensures improved relation-aware representation, leading to more accurate classification. Our method demonstrates state-of-the-art performance on the CPM-RadPath2020 challenge dataset. Please see Chapter 4 for detailed explanation of the work.

Brain cancer has a very high mortality rate, with gliomas being the most common and deadly malignant brain tumors. Recent biological studies have shown that a comprehensive analysis of biomarkers responsible for genetic mutations in gliomas can lead to better prognosis and treatment plans for patients. We simultaneously predict five crucial genetic markers—IDH, 1p/19q codeletion status, ATRX, MGMT, and TERT, using deep learning on Whole Slide Images. Our deep learning solution features a novel composite loss function that combines individual, pairwise, and group-wise traits of these biomarkers. Multi-label weighted cross-entropy loss captures individual characteristics, a conditional probability loss models pairwise behavior, and a spectral graph loss addresses group properties. Extensive experiments and an ablation study demonstrate the effectiveness of our approach, achieving state-of-the-art prediction performance as described in Chapter 5.

In order to detect early signs of Alzheimer’s disease and to monitor the disease progression, automated classification of Alzheimer’s disease is essential. We perform a novel task of directly classifying four different stages, namely, Alzheimer’s Disease, Healthy, Early Mild Cognitive Impairment (EMCI), and Late Mild Cognitive Impairment (LMCI), using 3D DTI data. The DTI modality offers information regarding brain structure through measurements of Fractional Anisotropy (FA) and Mean Diffusivity (MD), as well as Echo Planar Imaging (EPI) intensities. We train CNNs, especially, VoxCNNs individually on different types of such data, like FA values, MD values, and EPI intensities from 3D DTI scan volumes. Furthermore, we input the mean FA and MD values for each specific brain region, which are calculated based on the Colin27 brain atlas, into a random forest classifier (RFC). These four models, consisting of three separately trained VoxCNNs and one RFC, are used independently to solve the four-class classification problem mentioned above. The

individual classification findings are combined at the decision level using a modulated rank averaging technique, resulting in a high classification accuracy. The proposed technique has been convincingly demonstrated to be effective through comprehensive experimentation on the publicly available ADNI database and with state-of-the-art comparisons, details of which is available in Chapter 6.

Hippocampus (HC) segmentation is crucial for diagnosing neurodegenerative diseases such as Alzheimer’s, Parkinson’s, and common neurological disorders like epilepsy. Studying HC atrophy can provide biomarkers and improve techniques for detecting and predicting these illnesses, making segmentation vital in neuro-radiology. While manual segmentation is highly accurate and considered the gold standard, it is extremely time-consuming and labor-intensive. This has led to the development of efficient automation techniques. In this work, we propose a 3D HC segmentation solution from MRI data using a shape-driven loss function and attention UNet. We develop a Histogram of Oriented Gradients (HOG) based formulation to extract shape features and suggest a pooling technique as an alternative to histogram calculation for HOG, please see Chapter 7 for the methodology in details. This addresses the issue of histograms being non-derivable, which makes error calculation from histograms unsuitable for backpropagation in deep learning models. Our model’s performance is validated on two publicly available datasets, HarP and Kulaga-Yoskovitz (KY), demonstrating superior segmentation accuracy compared to the attention UNet model with only Dice loss and other state-of-the-art approaches.

Examining the hippocampus (HC) subfields is crucial for detecting early signs of brain abnormalities. However, segmenting these subfields is challenging due to their complexity and the need for manually annotated high-resolution magnetic resonance images. Building on our previous HC segmentation work, we propose an innovative deep graph cut approach, enhanced by shape information, for automatic segmentation of HC subfields. Our method incorporates a deep-learned shape term into the graph cut energy function. Additionally, we designed a modified $\alpha - \beta$ swap technique that leverages deep learning to improve the execution time of the multi-class segmentation algorithm. The detailed description of the work can be

found in Chapter 7. Our solution outperforms several state-of-the-art methods on the publicly available KY dataset.

To summarize our contributions, we highlight the main aspect of each contribution below-

1. We have developed a superior 3D brain tumor segmentation method by fusing Graph Cut and Deep Learning techniques on MRI scans [83].
2. We have created an innovative multi-modal classification model for classifying Glioblastoma, Astrocytoma, and Oligodendroglioma brain tumors with high accuracy [85].
3. We have made predictions of glioma bio-markers using a model combining genetic data with Whole Slide Images, targeting IDH, 1p/19q codeletion, ATRX, TERT, and MGMT.
4. We have proposed a state-of-the-art classification model for simultaneously classifying four stages of Alzheimer’s Disease using from DTI data [86].
5. We have constructed a novel loss function for deep learning-based hippocampus segmentation [87] using MRI scans. We have further extended the solution to multi-class, improving diagnosis and treatment of diseases like Alzheimer’s Disease, Dementia, and Epilepsy.

Chapter 2

Theoretical Foundations

This chapter explores classical computer vision models, machine learning models, and deep learning models foundational to this thesis. Key concepts discussed include graph cuts for image segmentation, the Random Forest algorithm, and advanced neural network architectures such as Convolutional Neural Networks, DenseNet, and UNet, along with graph-based neural networks like Graph Convolutional Networks.

2.1 Classical Computer Vision Models

We first discuss about some classical computer vision models used in this thesis.

2.1.1 Graph Cut

Firstly, we define the fundamental terminology used in this context. Consider a graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$, which comprises a set of nodes \mathcal{V} and a set of directed edges \mathcal{E} connecting these nodes. The node set $\mathcal{V} = \{s, t\} \cup \mathcal{P}$ includes two distinguished terminal nodes, namely the source s and the sink t , alongside a set of non-terminal nodes denoted as \mathcal{P} . Figure 2.1 presents a simple illustration of a graph containing terminals s and t . Such N-dimensional grids are commonly utilized in vision and graphics applications.

Each edge within the graph is associated with a non-negative weight or cost $w(p, q)$. The cost associated with a directed edge (p, q) may differ from that of the reverse edge (q, p) . An edge is termed a *t-link* if it connects a non-terminal node

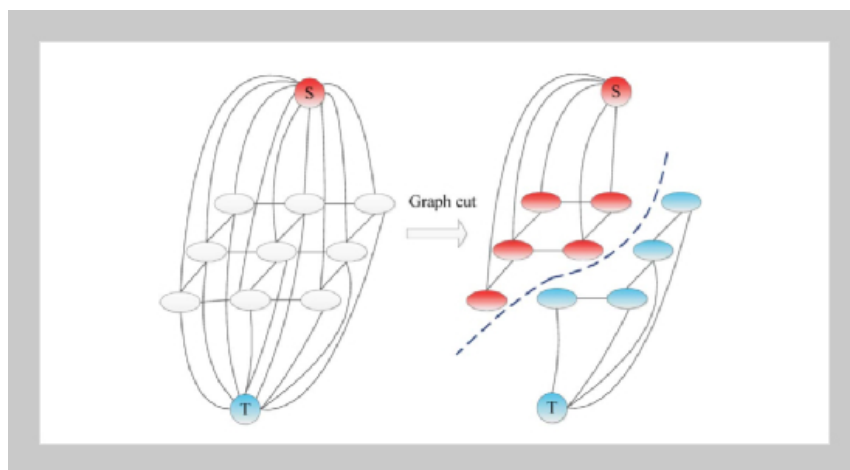


Figure 2.1: Graph construction for a 3×3 image. The dotted line denotes a cut. [4]

in \mathcal{P} with a terminal node. Conversely, an edge is termed an *n-link* if it connects two non-terminal nodes. The collection of all (directed) n-links is represented by \mathcal{N} . The set of all edges \mathcal{E} in the graph comprises the n-links in \mathcal{N} and the t-links $\{(s, p), (p, t)\}$ for non-terminal nodes $p \in \mathcal{P}$. As shown in Figure 2.1, the t-links are the edges attached to the source S and the sink T , whereas, the n-links are the edges connecting any two gray pixels.

(A) The Min-Cut and Max-Flow Problem

An *s/t* cut \mathcal{C} (often simply referred to as a cut) is a partition of the graph's nodes into two disjoint subsets \mathcal{S} and \mathcal{T} such that the source s belongs to \mathcal{S} and the sink t belongs to \mathcal{T} . Figure 2.1 illustrates an example of such a cut. The cost of a cut $\mathcal{C} = \{\mathcal{S}, \mathcal{T}\}$ is determined by summing the weights of the “boundary” edges (p, q) where $p \in \mathcal{S}$ and $q \in \mathcal{T}$. When (p, q) is a boundary edge, it is sometimes stated that the cut \mathcal{C} severs the edge (p, q) . The minimum cut problem seeks to identify a cut with the minimum cost among all possible cuts.

A pivotal result in combinatorial optimization reveals that the minimum *s/t* cut problem can be resolved by determining the maximum flow from the source s to the sink t . Informally, the maximum flow represents the greatest “amount of water” that can be transported from the source to the sink, with the graph edges interpreted as directed “pipes” having capacities equivalent to edge weights. The

Ford-Fulkerson theorem [88] asserts that a maximum flow from s to t saturates a set of edges within the graph, thereby dividing the nodes into two disjoint subsets $\{\mathcal{S}, \mathcal{T}\}$ corresponding to a minimum cut. Consequently, the min-cut and max-flow problems are equivalent, with the value of the maximum flow equating to the cost of the minimum cut.

2.1.2 Graph Cuts for Solving the Max-Flow Min-Cut Problem

Graph cuts address the max-flow min-cut problem by constructing a flow network and identifying the flow configuration that maximizes the flow from the source s to the sink t .

(A) Energy Function in Graph Cuts

The energy function in graph cuts is typically used to formulate and solve segmentation problems. The energy function $E(f)$ generally consists of two terms: the data term and the smoothness term.

$$E(f) = E_{\text{data}}(f) + E_{\text{smooth}}(f) \quad (2.1)$$

(B) Data Term:

The data term $E_{\text{data}}(f)$ measures how well the labeling f fits the observed data. It is often defined as:

$$E_{\text{data}}(f) = \sum_{p \in \mathcal{P}} D_p(f_p) \quad (2.2)$$

where $D_p(f_p)$ is the penalty for assigning label f_p to pixel p .

(C) Smoothness Term:

The smoothness term $E_{\text{smooth}}(f)$ encourages spatially coherent labelings. It is typically defined as:

$$E_{\text{smooth}}(f) = \sum_{(p,q) \in \mathcal{N}} V_{pq}(f_p, f_q) \quad (2.3)$$

where $V_{pq}(f_p, f_q)$ is the penalty for assigning labels f_p and f_q to neighboring pixels p and q .

(D) Seeding:

Seeding is a crucial step in graph cuts for image segmentation. It involves manually or automatically marking certain pixels in the image as belonging to the foreground (object of interest) or the background. These marked pixels are referred to as seeds and is used to set the data term in the energy function more accurately. Pixels marked as foreground seeds are strongly biased towards being labeled as the foreground, and similarly, background seeds are biased towards being labeled as the background. This is achieved by assigning very high or very low data penalties $D_p(f_p)$ to these seeded pixels, ensuring they influence the segmentation result significantly.

(E) Minimizing the energy

In image segmentation, the goal is to partition the image into regions that correspond to different objects or textures. The graph cut method can be used to find an optimal segmentation by minimizing the energy function defined above. By finding the minimum cut, as shown in an example graph in Fig. 2.2, we can obtain the optimal labeling f that minimizes the energy function $E(f)$. This minimum cut corresponds to the segmentation that best fits the data while maintaining spatial coherence. The algorithm for graph cut is shown in Algorithm. 2.1.

Algorithm 2.1: Graph Cut Algorithm

Input: Graph $G = (V, E)$ where V is the set of vertices and E is the set of edges.

Input: Source node $s \in V$ and sink node $t \in V$.

Input: Energy function $c : E \rightarrow \mathbb{R}^+$.

Output: A minimum cut (S, T) that separates the source s and sink t .

- 1 **Initialize** $S = \{s\}$ and $T = V \setminus S$;
- 2 **while** *there exists an augmenting path P from s to t in the residual graph G_f* **do**
- 3 Find the augmenting path P using Breadth First Search (BFS);
- 4 Determine the bottleneck capacity $\Delta = \min\{c_f(u, v) \mid (u, v) \in P\}$;
- 5 **foreach** *edge (u, v) in the path P* **do**
- 6 Subtract Δ from the forward edge capacity: $c_f(u, v) = c_f(u, v) - \Delta$;
- 7 Add Δ to the reverse edge capacity: $c_f(v, u) = c_f(v, u) + \Delta$;
- 8 Identify the set S of all vertices reachable from s in the residual graph G_f ;
- 9 Let $T = V \setminus S$;
- 10 **return** *The minimum cut is the set of edges (u, v) where $u \in S$ and $v \in T$;*

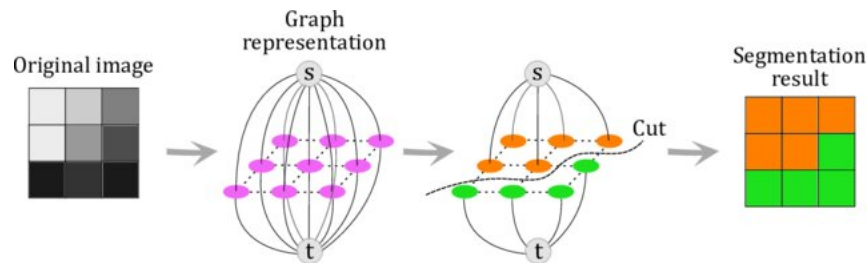


Figure 2.2: An example of segmentation a 3×3 image with graph cut. [5]



Figure 2.3: Segmentation of the people and bell in the foreground from the background area using graph cut with seeding [6]

(F) Time Complexity

The time complexity of the graph cut algorithm using Ford-Fulkerson's method depends on the maximum flow in the network and the method used to find augmenting paths. In the worst case, Ford-Fulkerson's algorithm has a time complexity of $O(E \cdot f^*)$, where E is the number of edges in the graph and f^* is the value of the maximum flow. This is because the algorithm may require f^* iterations, each involving a depth-first search (DFS) or breadth-first search (BFS) to find an augmenting path, which takes $O(E)$ time. When using BFS (Edmonds-Karp implementation), the complexity improves to $O(V \cdot E^2)$, where V is the number of vertices, as each augmenting path can be found in $O(E)$ time and there are at most $O(V \cdot E)$ augmenting paths [88].

In summary, graph cuts provide a powerful framework for solving the max-flow min-cut problem and optimizing energy functions for tasks such as image segmentation. Fig. 2.3 shows an example of an actual image segmented into foreground and background by using seeds. The combination of efficient algorithms, intuitive geometric interpretations, and the ability to incorporate user-provided seeds makes graph cuts a widely used tool in computer vision and graphics.

2.1.3 Multilabel Graph Cuts

While the standard graph cut method is inherently binary, many real-world applications require labeling pixels with more than two labels. This is where multilabel graph cuts come into play, extending the binary graph cut framework to handle multiple labels effectively.

(A) Formulation of the Multilabel Problem:

Given a set of sites (pixels) \mathcal{P} and a set of labels \mathcal{L} where $|\mathcal{L}| > 2$, the goal is to find a labeling $f : \mathcal{P} \rightarrow \mathcal{L}$ that minimizes an energy function of the form:

$$E(f) = \sum_{p \in \mathcal{P}} D_p(f_p) + \sum_{(p,q) \in \mathcal{N}} V_{pq}(f_p, f_q) \quad (2.4)$$

where:

- $D_p(f_p)$ is the data cost for assigning label f_p to pixel p .
- $V_{pq}(f_p, f_q)$ is the smoothness cost for assigning labels f_p and f_q to neighboring pixels p and q .

(B) Graph Construction for Multilabel Problems

The graph construction for multilabel problems involves creating additional nodes and edges to accommodate multiple labels. One common approach is to use the *alpha-expansion* and *alpha-beta swap* algorithms, which iteratively solve a series of binary graph cut problems to approximate the multilabel solution.

α -Expansion technique The alpha-expansion algorithm iteratively improves the labeling by allowing groups of pixels to change their label to a specified label α , while keeping other labels fixed. This process is repeated for all labels $\alpha \in \mathcal{L}$ until convergence. The key idea is to solve the following binary problem at each iteration:

$$E_{\text{exp}}(f, \alpha) = \sum_{p \in \mathcal{P}} D_p(f_p) + \sum_{(p,q) \in \mathcal{N}} V_{pq}(f_p, \alpha) \quad (2.5)$$

Following is the algorithm for α expansion.

Algorithm 2.2: α - expansion algorithm

Input: Initial labeling f , set of labels \mathcal{L} , energy function $E(f)$.

Output: Final labeling f that minimizes the energy function.

```

1 Initialize with an arbitrary labeling  $f$ ;
2 Set success := 0;
3 repeat
4   foreach label  $\alpha \in \mathcal{L}$  do
5     Find  $\hat{f} = \arg \min_{f'} E(f')$  among  $f'$  within one  $\alpha$ -expansion of  $f$ ;
6     if  $E(\hat{f}) < E(f)$  then
7       Set  $f := \hat{f}$ ;
8       Set success := 1;
9 until success = 0;
10 return Final labeling  $f$ ;

```

α - β Swap technique The alpha-beta swap algorithm improves the labeling by considering pairs of labels α and β . It allows pixels labeled α to switch to β and vice versa, aiming to find a better labeling configuration. The algorithm iteratively solves binary problems of the form:

$$E_{\text{swap}}(f, \alpha, \beta) = \sum_{p \in \mathcal{P}} D_p(f_p) + \sum_{(p,q) \in \mathcal{N}} V_{pq}(f_p, \beta) \quad (2.6)$$

Following is the algorithm for the same-

(C) Energy Minimization

Both the alpha-expansion and alpha-beta swap algorithms rely on solving binary subproblems using graph cuts. By iteratively applying these algorithms, it is possible to approximate the global minimum of the multilabel energy function.

(D) Applications of Multilabel Graph Cuts:

Multilabel graph cuts are widely used in various computer vision tasks, such as image segmentation, stereo vision, and texture synthesis. In image segmentation, for example, each label may correspond to a different object or region in the image,

Algorithm 2.3: $\alpha - \beta$ swap algorithm

Input: Initial labeling f , set of labels \mathcal{L} , energy function $E(f)$.**Output:** Final labeling f that minimizes the energy function.

```

1 Initialize with an arbitrary labeling  $f$ ;
2 Set success := 0;
3 repeat
4   foreach pair of labels  $\{\alpha, \beta\} \subset \mathcal{L}$  do
5     Find  $\hat{f} = \arg \min_{f'} E(f')$  among  $f'$  within one  $\alpha - \beta$  swap of  $f$ ;
6     if  $E(\hat{f}) < E(f)$  then
7       Set  $f := \hat{f}$ ;
8       Set success := 1;
9 until success = 0;
10 return Final labeling  $f$ ;

```

allowing for more complex and detailed segmentation results compared to binary methods.

In summary, multilabel graph cuts extend the powerful binary graph cut framework to handle multiple labels, providing effective solutions for complex labeling problems in computer vision. By iteratively solving binary subproblems, multilabel graph cuts achieve near-optimal results for various applications.

2.2 Machine Learning Models

Machine Learning (ML) is a subset of artificial intelligence (AI) focused on developing algorithms that enable computers to learn from and make decisions based on data. ML models identify patterns and relationships within data, allowing them to make predictions or decisions without explicit programming for specific tasks. The advantages of ML include task automation, improved accuracy, scalability, and adaptability to new and changing data. Its popularity has surged due to the increased availability of data, advancements in algorithms, enhanced computational power, and its versatility in application. ML is applied across various industries such as healthcare for diagnosing diseases and personalizing treatments, finance for fraud detection and risk management, retail for customer segmentation and demand forecasting. We discuss a popular ML model that we used in our research work.

2.2.1 Random Forest

Random Forest [89] is a highly regarded machine learning algorithm that utilizes supervised learning techniques. It exhibits considerable versatility, making it suitable for both classification and regression tasks. The algorithm is based on the concept of ensemble learning, which combines multiple classifiers to tackle complex problems and improve the overall performance of the model.

Fundamentally, a Random Forest consists of numerous decision trees, each of which is trained on different subsets of the dataset. By averaging the predictions generated by these individual trees, the algorithm enhances the accuracy of the model. Rather than depending on a single decision tree, the Random Forest aggregates the outcomes from all trees and determines the final output through a majority vote of their predictions.

(A) Working of the Random Forest Algorithm

The Random Forest Algorithm operates in two primary phases: building the forest (comprising N decision trees) and making predictions using these trees. The following steps outline this process:

1. **Random Sampling:** Randomly select M data points from the training set.
2. **Tree Construction:** Construct decision trees using these subsets of data.
3. **Prediction:** Each decision tree produces a prediction.
4. **Aggregation:** For classification, use majority voting; for regression, compute the average of predictions.

For instance, consider a scenario where we have a dataset comprising multiple images of various fruits. This dataset is provided to the Random Forest Classifier. Each decision tree within the forest is assigned a subset of the dataset for analysis. During the training phase, each decision tree produces a prediction outcome. When a new data point is introduced, the Random Forest classifier determines the final prediction by considering the majority of the outcomes generated by the individual trees. See

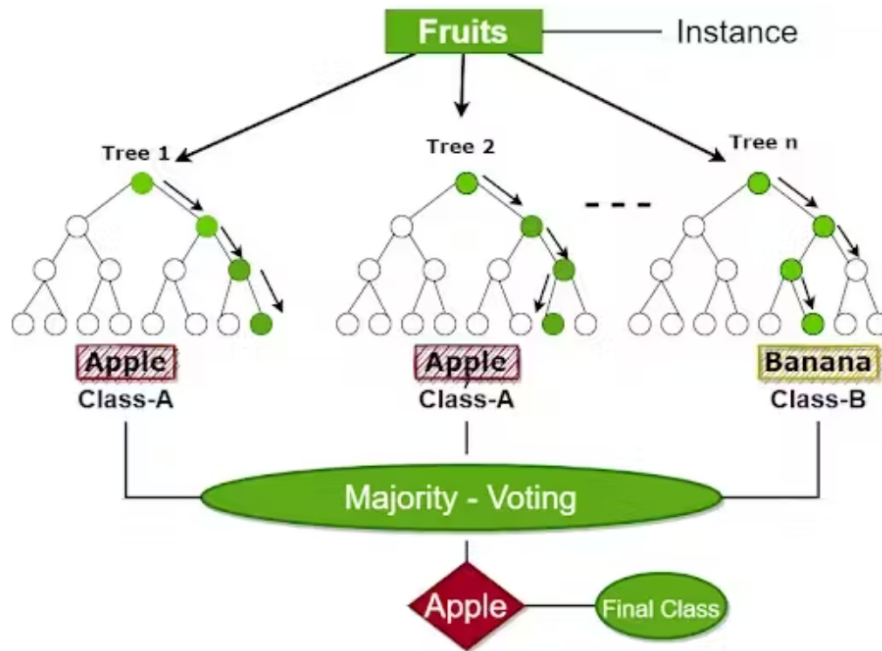


Figure 2.4: Depiction of how the prediction of individual decision tree is used in predicting the final label in a random forest

Fig. 2.4 for reference. The algorithm of random forest is given in Algorithm. 2.4. Mathematically, a Random Forest is an ensemble of N decision trees, denoted as $\{T_1, T_2, \dots, T_N\}$. Given an input feature vector x , the final prediction \hat{y} is obtained by aggregating the individual predictions of all decision trees:

$$\hat{y} = f(T_1(x), T_2(x), \dots, T_N(x)) \quad (2.7)$$

The aggregation function $f(\cdot)$ differs depending on the type of task:

Classification: For classification, the final prediction is determined by majority voting among the predictions of the N trees. Specifically, the predicted class \hat{y} is given by:

$$\hat{y} = \arg \max_k \sum_{i=1}^N I(T_i(x) = k) \quad (2.8)$$

where: - k represents a possible class label. - $T_i(x)$ is the class predicted by the i -th decision tree for input x . - $I(T_i(x) = k)$ is an indicator function that equals 1

Algorithm 2.4: Random Forest Algorithm

Input: Training dataset D with n samples and m features, number of trees N , number of features k to consider for splitting at each node.

Output: A random forest model consisting of N decision trees.

```

1 for  $i := 1$  to  $N$  do
2   Generate a bootstrap sample  $D_i$  from the training dataset  $D$  by
   sampling with replacement;
3   Grow a decision tree  $T_i$  on  $D_i$  as follows:
   • while the stopping criterion is not met do
     Randomly select  $k$  features from the  $m$  available features;
     Choose the best split among the  $k$  features based on a criterion
     (e.g., Gini impurity or information gain);
     Split the node into two child nodes;
   Add the tree  $T_i$  to the forest;
4 To make predictions on a new sample  $x$ :
   • Aggregate the predictions from all the trees  $\{T_1, T_2, \dots, T_N\}$  in the forest
     (e.g., by majority vote for classification or averaging for regression);

return The aggregated prediction for the new sample;

```

if the i -th tree predicts class k , and 0 otherwise. - The class with the highest vote count is selected as the final prediction.

Regression: For regression tasks, the final prediction is obtained by averaging the outputs of all decision trees:

$$\hat{y} = \frac{1}{N} \sum_{i=1}^N T_i(x) \quad (2.9)$$

where: - $T_i(x)$ is the numerical output predicted by the i -th tree for input x .
 - The arithmetic mean ensures that the final prediction is a smoothed estimate, reducing variance compared to individual tree predictions.

This ensemble approach helps improve model stability and generalization by reducing overfitting, which is a common issue in single decision trees.

(B) Advantages of Random Forests

Random Forest Algorithm has the following advantages-

- **Diversity:** Each tree is built using a different subset of features, ensuring diversity.
- **No Overfitting:** Random Forests are less prone to over-fitting since they aggregate results from multiple trees.
- **Dimensionality:** The algorithm effectively handles high-dimensional data by considering only a subset of features for each tree.
- **Parallelization:** Trees are constructed independently, allowing for parallel processing.
- **Train-Test Split:** There is no need for a separate train-test split as each tree inherently samples different subsets of data.
- **Stability:** The algorithm provides stable predictions through majority voting or averaging.

(C) Time complexity of Random Forests

The time complexity of Random Forest is given as-

$$T_{rfc} = O(R \cdot p \cdot q \cdot \log(p)) \quad (2.10)$$

where R is the number of trees, p is the number of samples, and q is the number of features considered in each split.

Random Forest is a highly effective and versatile machine learning algorithm, widely used across various industries for its robustness and accuracy. Despite its complexity and longer training time compared to simpler models like decision trees, it remains a popular choice due to its ability to handle diverse data types and provide stable predictions.

2.2.2 Artificial Neural Networks

As the name suggest, the design of artificial neural networks (ANN) are inspired from the structure of biological neurons of human brain. A normal human brain is

approximately composed of around 85 billion neurons, which are responsible for controlling human actions and behaviors. As shown in the figure 2.5 a neuron react to other neurons to transmit information based on the membrane potential that gradually increases with the reception of some excitatory input. If the membrane potential reaches a certain level over a specific threshold, an action potential is triggered that propagates along the axon to the post-synaptic terminals. In a mathematical sense, an artificial neuron is modeled as a function that first computes a weighted sum of the input vector x based on weight vector w and offset by a bias term b . This result is usually passed through a non-linear “activation function” as:

$$y = \sigma \left(\sum_i w_i x_i + b \right) = \sigma (w^T x + b) \quad (2.11)$$

(A) Activation Functions

There are several activation functions as shown in figure 2.6. Initially, activations are achieved through simple thresholding, but such models are difficult to train due to non-derivable nature. Function similar to the nature of sigmoid function like hyperbolic tangent or the logistic function are most commonly adopted for a while before Rectified Linear Unit (ReLU) was introduced as activation function. Each neuron performs simple computations but can result in complex output when combined with several other neurons arranged in a specific pattern. Theoretically, neurons can be organized quite arbitrarily, but usually they are arranged as directed acyclic graphs, *i.e.*, input to a neuron does not depend on its output. Such form of arrangements are also widely known as “feed-forward” neural network as the activations only propagates in forward direction.

(B) Multi-Layer Perceptrons (MLP)

The most popular feed-forward neural networks are multi-layer perceptrons (see figure 2.7). A classic MLP contains one hidden and one output layer in addition to an input layer. When each neuron in a layer is connected to all other neurons

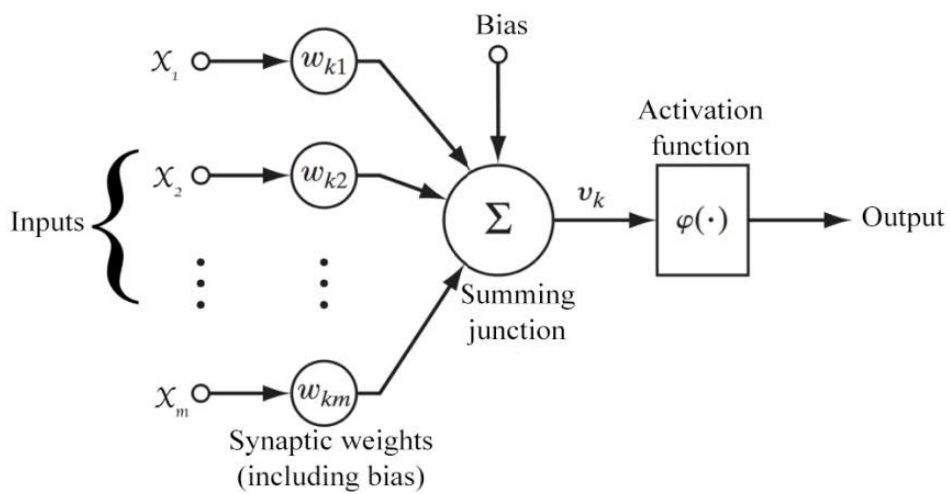
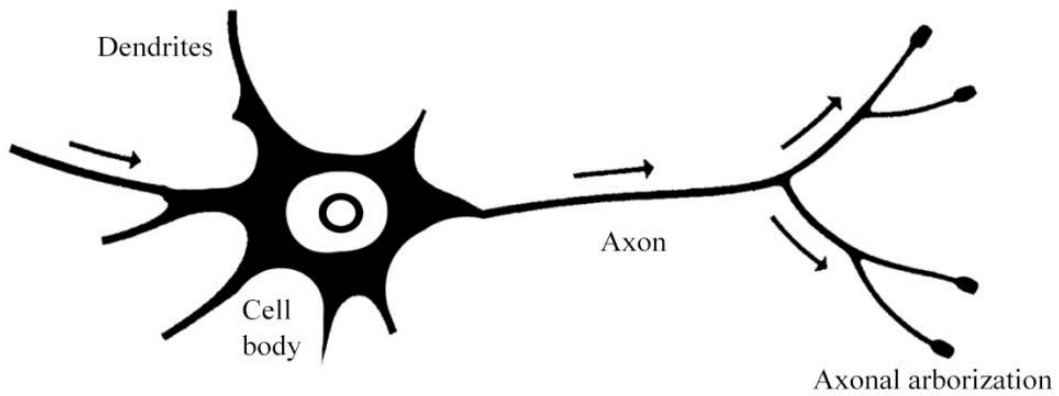
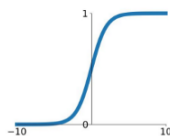


Figure 2.5: Biological Neuron Vs Artificial Neuron [7]

Activation Functions

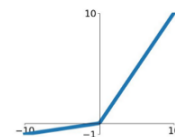
Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



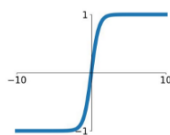
Leaky ReLU

$$\max(0.1x, x)$$



tanh

$$\tanh(x)$$

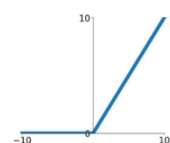


Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

ReLU

$$\max(0, x)$$



ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$

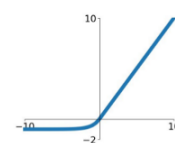


Figure 2.6: Different Neural Activation functions

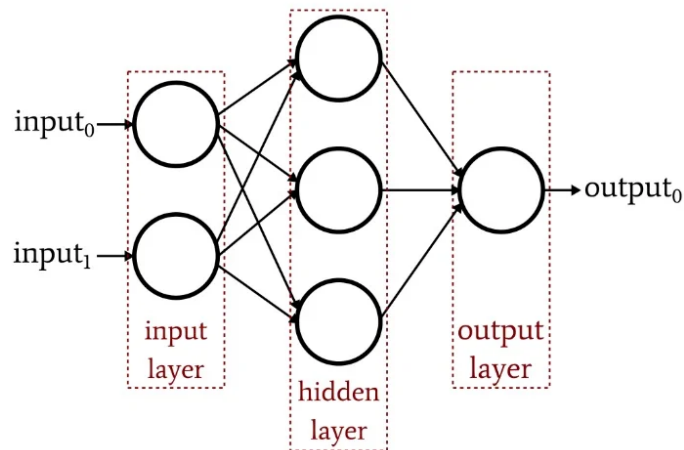


Figure 2.7: A typical illustration of a MLP

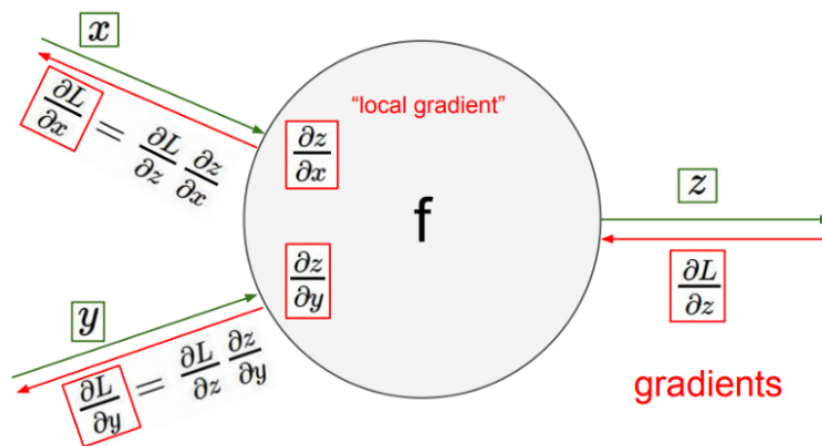


Figure 2.8: Illustration of back-propagation through a neuron.

in the preceding layer also known as fully connected layer (see 2.7). The basic idea behind such design is that several simple elementary functions can be combined to approximate a complex function simply by tuning the parameters of the network. In this context, the universal approximation theorem [90] states that only a single hidden layer based feed forward network consisting of a finite set of neurons can approximate any continuous function under with assumption on activation function.

(C) Backpropagation

The backpropagation algorithm [91] is the most widely used approach for training feed-forward neural networks. It is a specially designed version of gradient descent

optimization in the context of updating/learning neural networks. The term “back-propagation” used as an abbreviation of “backward propagation of errors”. During neural training, backpropagation is followed immediately after computation of feed-forward activations to compute the gradients of errors for updating the weights. Let L be the loss/cost function computed between outputs and the desired targets for a training sample batch (see figure 2.8). Let the gradient received by the neuron from the preceding layer be represented by partial derivative $\partial L/\partial z$. We further require storing the local gradients ($\partial z/\partial x$ and $\partial z/\partial y$), activation (z) and inputs (x and y) for each neuron in a memory for later calculations. These gradients are further required to calculate gradients with respect to the inputs x and y as $\partial L/\partial x$ and $\partial L/\partial y$ respectively. Also, the gradient of loss L with respect to the weight w can be computed as $\partial L/\partial w$ which can be further used to compute the updated weight based on gradient descent given a learning rate α as:

$$\Delta w = -\alpha \frac{\partial L}{\partial w} \quad (2.12)$$

A momentum term can also be further incorporated based on the weight of previous update scaled by a factor η for faster convergence as

$$\Delta^t = -\alpha \frac{\partial L}{\partial w^t} + \eta w^{t-1} \quad (2.13)$$

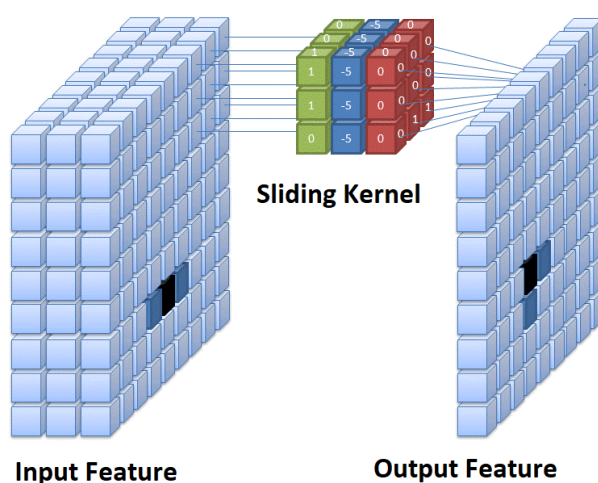


Figure 2.9: Illustration of Convolutional Layer

2.3 Deep Learning Models

Deep Neural Networks (DNNs) are a subset of ANNs with multiple layers between the input and output layers. This architecture enables DNNs to learn complex data patterns, surpassing the capabilities of traditional machine learning models. By incrementally processing information through successive layers, DNNs can extract increasingly abstract features. A prime example of this is the convolutional layer, especially designed to process grid-like data such as images. This layer applies filters to input data to extract essential features, laying the groundwork for advanced computer vision tasks.

2.3.1 Convolutional Layer

A seminal breakthrough in the neurophysiology of the visual cortex emerged in 1962 with Hubel and Wiesel's discovery of orientation selectivity and columnar organization [92]. Employing microelectrode recordings from single neurons in the primary visual cortex (V1) of anesthetized cats, they initially encountered challenges in eliciting responses to projected light and dark dots. Serendipitously, the slide projector's margin induced neuronal activation, prompting further investigation with light and dark bars of varying orientations. This led to the characterization of two cell types: simple cells, exhibiting selective responses to specific bar orientations, and complex cells, responding to a broader range of orientations while maintaining a preference. Crucially, these cells were found to be hierarchically organized, receiving input from simpler cells and constituting the foundational elements of visual processing. Fukushima's Neocognitron [93], a precursor to contemporary convolutional neural networks, was directly inspired by the groundbreaking research of Hubel and Wiesel. Working at the NHK Science & Technology Research Laboratories, Kunihiko Fukushima translated the neurophysiological findings of simple and complex cells into a multilayered neural network architecture. This pioneering work laid the foundation for the development of artificial systems capable of processing visual information in a manner analogous to the mammalian visual cortex. An image usually consists of several hundreds of pixels that needs to be processed in any computer

vision task. Utilizing only fully-connected networks like MLP would exponentially increase the number of parameters at the early stages of a neural network leading to higher risk of overfitting. Convolutional layer was first to successfully address this issues of MLP for images classification tasks also known as Convolutional neural network (CNN). Instead of connecting every pixel of the input feature map to every pixel of next hidden layer, a finite set of filter kernels are used to convolve over the feature to obtain the next feature map. These kernels are of fixed size and also require much less parameters than fully-connected layer. The two main aspects of a CNN layer are a) local receptive fields where a hidden node is only connected to a smaller region from previous layer and b) shared weights for all mappings between local receptive field from previous layer with subsequent hidden node (see Fig. 2.9). This two sets of restrictions dramatically reduce the required network parameters. The convolution operation can be mathematically expressed as:

$$y = \sigma(W * X + b) \quad (2.14)$$

where, X and W are the input and weight matrix respectively. Here, b is a bias term with activation function σ and $*$ denotes discrete convolution operation. For 2D input image X , the convolution operation can be element-wise expressed as:

$$(W * X)(i, j) = \sum_m \sum_n X(m, n)W(i - m, j - n) \quad (2.15)$$

The output matrix of convolutional layer is obtained by sliding the weight matrix W over the input X followed by a dot product (see Fig. 2.9). Here a learned weight matrix W is composed of several filter kernels that fire up in response to a specific feature in input X . This response may occur in form of edges, color or pattern that mostly correlates with the filter characteristics. This filter kernels are automatically learned during back propagation from the data itself during training.

Time complexity of convolutional layer

The time complexity for convolutional layers is given by [94]-

$$T_{\text{conv}} = \sum_{n=1}^d k_{n-1} \cdot s_n^2 \cdot f_n \cdot l_n^2 \cdot r_1 \cdot b_1 \quad (2.16)$$

where:

- d is the number of convolutional layers.
- l_n is the length of the output feature map in the n th layer.
- f_n is the number of filters in the n th layer.
- s_n is the filter size in the n th layer.
- k_{n-1} is the number of input channels.
- r_1 is the learning rate.
- b_1 is the batch size.

For a convolutional layer in 3D, the time complexity can be written as-

$$T_{\text{conv}}^{3D} = \sum_{n=1}^d k_{n-1} \cdot s_n^3 \cdot f_n \cdot l_n^3 \cdot r_1 \cdot b_1 \quad (2.17)$$

where symbols have their usual meanings as given in the above Eq. 2.16.

2.3.2 Fully Connected Layer

A Fully Connected (FC) layer is a fundamental component of deep learning models, particularly in feedforward neural networks and convolutional neural networks (CNNs). In an FC layer, each neuron is connected to every neuron in the previous layer, allowing the network to learn complex hierarchical representations. This layer performs a linear transformation followed by a non-linear activation function, enabling the model to capture intricate patterns in data. Despite its effectiveness, fully connected layers tend to have a large number of parameters, which can lead to high

computational costs and overfitting, particularly when used with high-dimensional input data. In modern deep learning architectures, FC layers are often used in the final classification stage after feature extraction layers, such as convolutional or recurrent layers, to aggregate learned features into class predictions [95].

Mathematically, a fully connected layer can be represented as follows: Given an input vector $x \in \mathbb{R}^n$, the output of the FC layer is computed as -

$$y = f(Wx + b), \quad (2.18)$$

where $W \in \mathbb{R}^{m \times n}$ is the weight matrix, $b \in \mathbb{R}^m$ is the bias vector, and f is a non-linear activation function (e.g., ReLU, Sigmoid, or Softmax). The weight matrix W learns the mapping between input features and the output neurons, while the bias vector b allows for additional flexibility in learning shifts in data distribution. During training, these parameters are optimized using backpropagation and gradient descent to minimize a chosen loss function. The computational complexity of an FC layer is $O(mn)$, making it expensive for large-scale models, which is why techniques like weight sharing (CNNs) or sparsity-inducing methods (pruning) are used to improve efficiency [95].

Time complexity of FC layer

The time complexity for a fully connected layer with 2D input is [94]-

$$T_{fc} = \sum_{l=1}^f C \cdot W \cdot H \cdot N \quad (2.19)$$

where:

- f is the number of fully connected layers.
- C is the number of input channels.
- W is the width of the input.
- H is the height of the input.
- N is the number of neurons.

The same for 3D input can be written as -

$$T_{fc}^{3D} = \sum_{l=1}^f C \cdot D \cdot W \cdot H \cdot N \quad (2.20)$$

where D is the depth of input and the other symbols have their usual meanings as given in Eq. 2.19.

2.3.3 Deep CNN Architectures

In this section, we discuss some popular CNN based deep neural networks that we have used in our work. We specifically chose these networks to suit our particular use-cases. For example, we have used a modified version of Alexnet in one of our works (described in Chapter 6) for learning to classify MRI images. In another work (Chapter 4), we had to extract deep features from images for which Densenet was a suitable choice [96]. For segmentation tasks (as done in Chapters 7, 3), Unet is the most popular deep learning model [97].

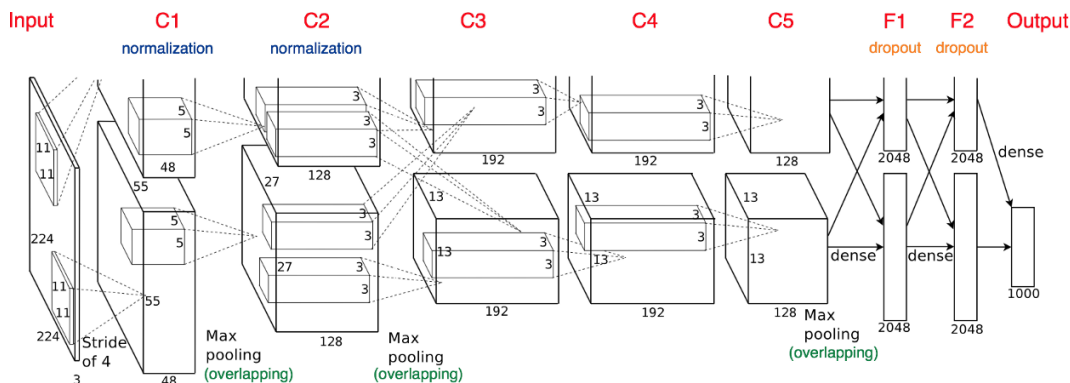


Figure 2.10: Architecture of Alexnet

(A) Alexnet

AlexNet [98] architecture gained much popularity by winning over ImageNet Large Scale Visual Recognition Competition (ILSVRC) in 2012 for image classification. The classification performance is achieved over a subset of ImageNet dataset consisting over 15 million high-resolution labeled images with roughly 22,000 categories. Out of these categories only 1000 categories are selected in the subset dataset and

for each selected categories only 1000 images are selected. AlexNet with 5 convolutional layer and 3 fully connected layer (see figure 2.10) have shown a significant performance gain over traditional approaches. In Alexnet, ReLU is used as activation function which help tackle the vanishing gradient and over-fitting problem to some extent which are persistent problem of sigmoid activation.

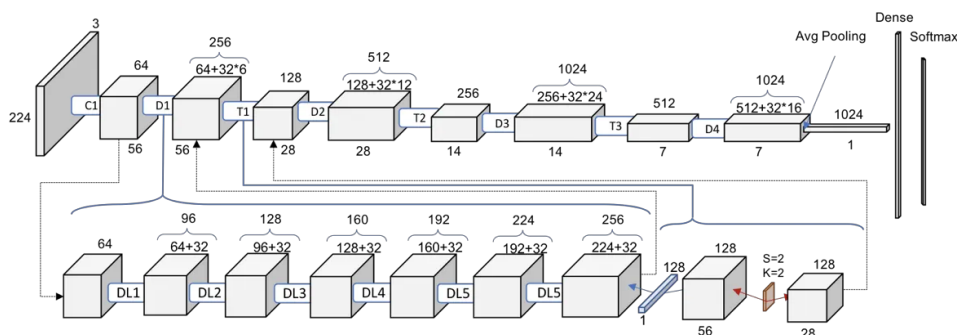


Figure 2.11: Architecture of DenseNet 121. Dx: Dense Block x. Tx: Transition Block x.

(B) DenseNet

DenseNet [32] is a convolutional neural network (CNN) architecture that establishes direct connections between each layer and all subsequent layers in a feed-forward manner. Unlike conventional architectures, which have L connections (one between each layer and the next), DenseNet features $\frac{L(L+1)}{2}$ direct connections. This unique structure mitigates the vanishing-gradient problem, enhances feature propagation, promotes feature reuse, and significantly reduces the number of parameters required. In DenseNet, each layer receives as input the feature maps from all preceding layers and transmits its own feature maps to all subsequent layers. DenseNet has been demonstrated to require fewer parameters than traditional CNNs while maintaining high performance on benchmark datasets. The architecture is composed of dense blocks, each containing several convolutional layers, and transition layers, which include batch normalization, convolution, and pooling layers that connect these dense blocks, as depicted in Fig. 2.11. In our work, we have employed DenseNet-121, where the dense blocks $D1$, $D2$, $D3$, and $D4$ consist of 6, 12, 24, and 16 layers, respectively. DenseNet has achieved competitive results on highly challenging image

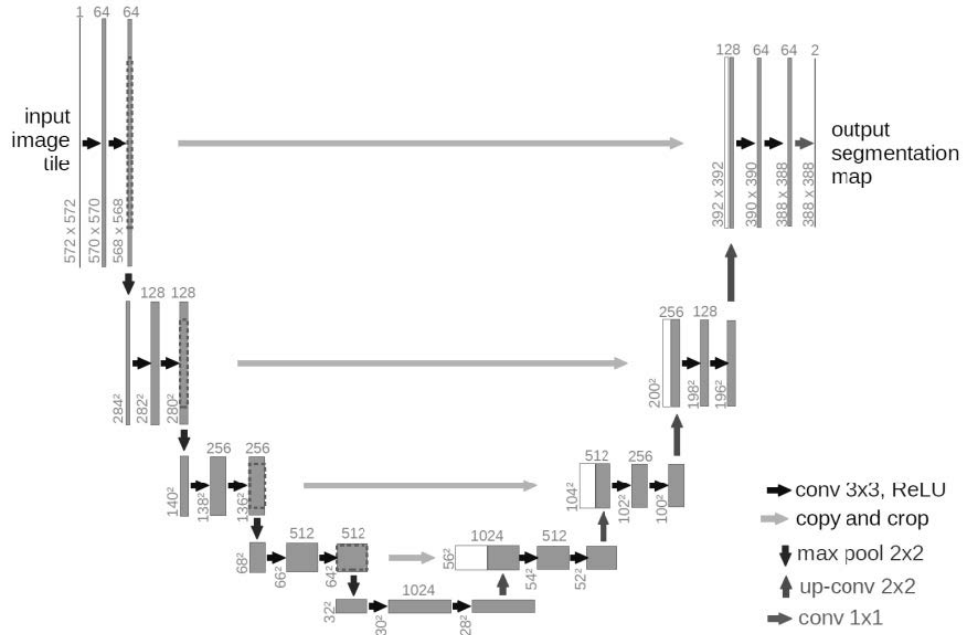


Figure 2.12: Architecture of U-Net

recognition tasks, underscoring its robustness and efficiency.

Using the equations described in Eq. 2.16 and Eq. 2.19, we can derive the time complexity for a 2D DenseNet-121 model having 120 convolutional layers and 1 fully connected layer as-

$$\begin{aligned}
 T_{D121}^{2D} &= T_{conv}^{2D} + T_{fc}^{2D} \\
 &= \sum_{n=1}^{120} k_{n-1} \cdot s_n^2 \cdot f_n \cdot l_n^2 \cdot r_1 \cdot b_1 + C \cdot D \cdot W \cdot H \cdot N
 \end{aligned} \tag{2.21}$$

where symbols have their usual meaning defined in Sec. 2.3.1 and 2.3.2.

(C) UNet

UNet [33], a CNN architecture, is specifically developed for biomedical image segmentation. It has garnered considerable recognition for its capability to deliver accurate and efficient segmentations with a relatively small amount of training data. The UNet architecture is characterized by its symmetrical design, comprising a contracting path (encoder) and an expansive path (decoder), forming a U-shaped structure, as illustrated in Figure 2.12.

The contracting path adheres to the conventional architecture of a convolutional

network. It consists of the repeated application of two 3x3 convolutions (unpadded), each followed by a rectified linear unit (ReLU) activation and a 2x2 max pooling operation with a stride of 2 for downsampling. With each downsampling step, the number of feature channels is doubled.

The expansive path, which is more intricate, is responsible for upsampling the feature maps back to the original image size. It involves the upsampling of the feature map, followed by a 2x2 convolution (referred to as "up-convolution") that reduces the number of feature channels by half. This is then concatenated with the corresponding cropped feature map from the contracting path, followed by two 3x3 convolutions, each succeeded by a ReLU activation. This symmetrical architecture enables the network to effectively capture both contextual information and precise localization.

A key feature of UNet is the skip connections that link the corresponding layers of the contracting and expansive paths. These connections concatenate the feature maps from the encoder to the decoder layers, ensuring that the detailed spatial information is retained and used effectively during the upsampling process. This design helps to preserve high-resolution features which are critical for precise segmentation tasks.

UNet has been successfully applied to various image segmentation tasks in medical imaging, such as the segmentation of neuronal structures in electron microscopic stacks, cell tracking in video microscopy, and many other applications requiring high accuracy in pixel-level classification.

Time Complexity Analysis We first calculate the time complexity of the encoder path. Each convolutional layer in the encoder has a time complexity:

$$T_{\text{enc}} = \sum_{n=1}^{2L} k_{n-1} \cdot s_n^3 \cdot f_n \cdot l_n^3 \cdot d_n \cdot r_1 \cdot b_1 \quad (2.22)$$

where L is the number of downsampling stages and the other symbols have their usual meanings as described in 2.17 Similarly, the decoder consists of transposed convolutions followed by standard convolutions. Hence, the time complexity of the

decoder can be written as -

$$T_{\text{dec}} = \sum_{n=1}^{2L} k_{n-1} \cdot s_n^3 \cdot f_n \cdot l_n^3 \cdot d_n \cdot r_1 \cdot b_1 \quad (2.23)$$

where L is the number of upsampling stages

The total time complexity of U-Net is given by-

$$\begin{aligned} T_{\text{UNet}} &= T_{\text{enc}} + T_{\text{dec}} \\ &= 2 \sum_{n=1}^{2L} k_{n-1} \cdot s_n^3 \cdot f_n \cdot l_n^3 \cdot r_1 \cdot b_1 \end{aligned} \quad (2.24)$$

Thus, the asymptotic complexity is-

$$T_{\text{UNet}} = O\left(\sum_{n=1}^{4L} k_{n-1} \cdot s_n^3 \cdot f_n \cdot l_n^3 \cdot r_1 \cdot b_1\right) \quad (2.25)$$

2.4 Graph based neural networks

Graph-Based Neural Networks, such as Graph Neural Networks, extend traditional neural networks to graph-structured data, enabling effective learning from relationships and interactions between nodes. Applications include node classification, graph classification, link prediction, and clustering, revolutionizing fields like social network analysis and bioinformatics. We first discuss the types of tasks that can be performed using Graph based Neural Networks.

2.4.1 Tasks

(A) Graph-Level tasks

Graph-level tasks involve predicting properties or labels associated with an entire graph structure. For instance, given a molecular graph, the objective might be to forecast its odor or its potential binding affinity to a disease-related receptor. This problem is analogous to image classification tasks, such as those encountered in MNIST and CIFAR datasets, where the goal is to assign a class label to a complete image. Similarly, in natural language processing, sentiment analysis mirrors this

concept, aiming to determine the overall sentiment of a sentence.

(B) Node-Level tasks

Node-level tasks pertain to the prediction of attributes or roles associated with individual nodes within a graph structure. A canonical example is the Zachary’s karate club dataset [99], a social network graph depicting members affiliated with one of two karate clubs following a leadership dispute. Nodes represent individuals, edges signify social interactions, and the task involves classifying members’ allegiance to either of the two club leaders. Proximity to the respective leaders is a significant predictor of this classification.

We have performed node level classification in our work as described in Chapter 4.

(C) Edge-Level tasks

Edge-level inference involves predicting relationships between entities within a graph. In the context of image scene understanding, this translates to identifying connections between objects beyond simple object detection. By framing this as an edge-level classification problem, deep learning models can be employed to determine the existence or nature of relationships between image objects represented as graph nodes. This approach enables the discovery of intricate dependencies between entities by constructing a fully connected graph and subsequently pruning edges based on predicted relationship strengths. We discuss some basic concepts of Graph Neural Networks below.

2.4.2 Graph Neural Networks

A Graph Neural Network (GNN) constitutes an optimizable transformation applied to all graph attributes—nodes, edges, and global context—while preserving inherent graph symmetries (permutation invariances). This study adopts the “message passing neural network” framework proposed by Gilmer *et al.* [100] and the Graph Nets architecture outlined by Battaglia *et al.* [101]. GNNs operate on a “graph-in, graph-out” paradigm, accepting a graph as input with information encoded within

its nodes, edges, and global context. These embeddings undergo iterative transformation without altering the graph's underlying connectivity. A GNN layer comprises independent multilayer perceptrons (MLPs) or equivalent differentiable models applied to each graph component: nodes, edges, and global context. The MLP processes each node vector, generating a learned node embedding. Similarly, edge and global context embeddings are learned. To enhance representational power, multiple GNN layers can be stacked sequentially. As shown in 2.13, a graph is the input, and each component (V, E, U) gets updated by a MLP to produce a new graph. Each function subscript indicates a separate function for a different graph attribute at the n -th layer of a GNN model. While the GNN does not modify the input graph's connectivity or number of feature vectors, it significantly updates the embedding space. The output graph retains the same adjacency list but incorporates refined node, edge, and global context embeddings resulting from the GNN's transformation process.

2.4.3 Graph Convolutional Networks

Graph Convolutional Networks (GCNs) [34] are a type of neural network designed to operate on graph-structured data. Traditional neural networks and convolutional neural networks (CNNs) are typically designed for data that reside on regular grids, such as sequences (1D) or images (2D). However, many real-world data sets, such as social networks, biological networks, and knowledge graphs, are naturally represented as graphs, which are more complex structures that consist of nodes and edges

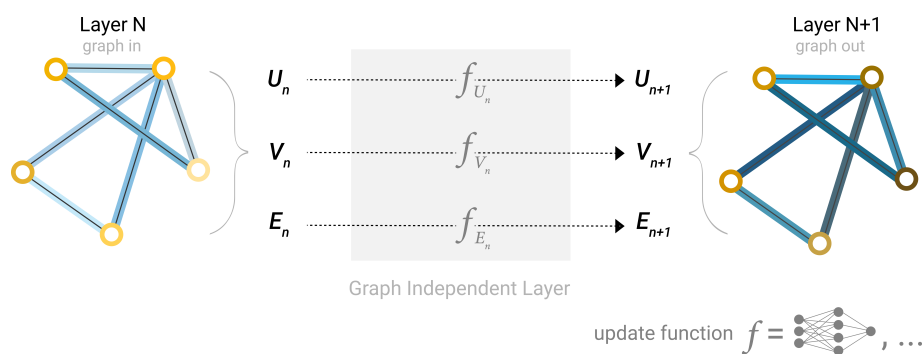


Figure 2.13: A single layer of a simple GNN.

without a fixed grid.

The concept of neural networks applied to graphs has been evolving for several years. Early approaches to graph neural networks (GNNs) were based on recurrent neural networks and were proposed to learn node representations by iteratively aggregating information from their neighbors. The Graph Neural Network model introduced by Scarselli *et al.* in 2009 [102] is one of the seminal works in this area. However, it was not until the introduction of GCNs by Kipf and Welling in 2017 [34] that the field saw significant advancements and widespread adoption.

GCNs extend the operation of traditional CNNs to graph-structured data by generalizing the convolution operation. The key idea is to redefine the convolutional operation to operate on the nodes of the graph, taking into account their connections as defined by the edges. Mathematically, the GCN layer can be expressed as:

$$H^{(l+1)} = \sigma \left(\tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2} H^{(l)} W^{(l)} \right) \quad (2.26)$$

where: - $H^{(l)}$ is the matrix of activations in the l -th layer, - $\tilde{A} = A + I$ is the adjacency matrix of the graph with added self-loops (where I is the identity matrix), the graph can be both weighted and unweighted. - \tilde{D} is the degree matrix of \tilde{A} , - $W^{(l)}$ is the weight matrix of the l -th layer, - σ is a non-linear activation function (e.g., ReLU).

The addition of self-loops ensures that each node's own features are included in the aggregation. The normalization by $\tilde{D}^{-1/2}$ ensures that the feature vectors are properly scaled.

GCNs have demonstrated powerful capabilities in a variety of applications:

- **Node Classification:** Assigning labels to nodes based on their features and the graph structure, e.g., classifying users in social networks or proteins in biological networks.
- **Graph Classification:** Predicting labels for entire graphs, e.g., determining the type of chemical compounds based on their molecular structure.
- **Link Prediction:** Predicting the existence of edges between nodes, useful in

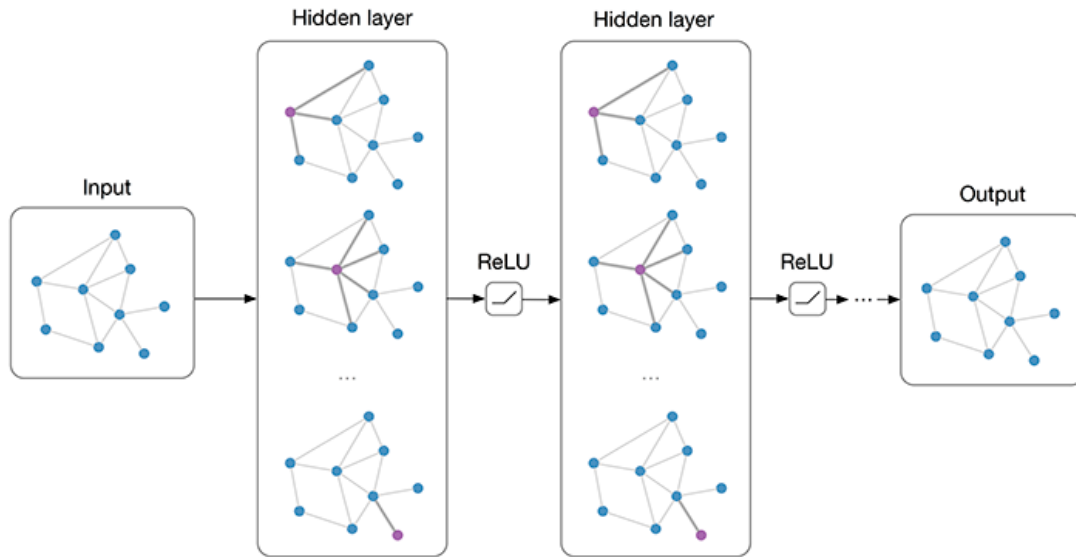


Figure 2.14: Architecture of a Multi-layer Graph Convolutional Network (GCN) with ReLU activation function

social network analysis and recommendation systems.

- Clustering: Grouping nodes into clusters, useful in community detection in networks.

The rise of GCNs has significantly impacted how we approach problems involving graph-structured data, providing a robust framework that leverages both the features of nodes and the intricate web of relationships between them.

Time Complexity of GCN

The time complexity for a single forward/backward pass of a GCN is $O(G_L \cdot G_N \cdot F^2 + G_L \cdot G_E \cdot F)$ [103], where G_L is the number of layers, G_N is the number of nodes, G_E is the number of edges and F is the number of features. So, the overall time complexity is the sum of the time complexity of 1 backward pass and 1 forward pass. Since, both the time complexities are same, the total time complexity a GCN can be represented as-

$$T_{gcn} = O(G_L \cdot G_N \cdot F^2 + G_L \cdot G_E \cdot F). \quad (2.27)$$

Chapter 3

Brain Tumor Segmentation

The development of computer vision techniques for brain tumor segmentation is vital for accurately delineating tumor boundaries, which is essential for precise diagnosis, treatment planning, and monitoring of brain cancer. Accurate segmentation allows for the identification of tumor characteristics, enabling personalized treatment strategies such as targeted radiotherapy and surgical planning that can improve patient outcomes. Moreover, these techniques enhance the consistency and reliability of tumor assessment, reducing the risks associated with manual interpretation and contributing to more effective and timely interventions in clinical practice.

3.1 Introduction

Formation of abnormal groups of cells inside or near the brain leads to brain tumor. These abnormal cells disrupt normal brain functions thereby leading to considerable degradation of the health of a patient. Brain image analysis [104] is considered imperative for detection of brain tumors as they are responsible for a large number of deaths world-wide. Currently radiologists use contrast to delineate tumor from surrounding brain parenchyma and for subsequent classification of the tumors as low grade (poor/minimal enhancement) to high grade (intense enhancement). The enhancement reflects the vascularity/neoangiogenesis within the tumor and break down of blood brain barrier. A challenge faced by the radiologist and the neurosurgeon is to demarcate the tumor margins when the lesion is poorly defined and is diffusely in-

filtrative. Recently, there have been concerns about the usage of Gadolinium and its deposition in the brain. Neuro-radiologists would prefer a shorter but accurate way of identifying tumors that can speed up the pre-operative diagnosis, maybe by using only a few essential MR sequences. In a centre with a large caseload, this would be extremely beneficial as it would free the MR scanning machine for more urgent scans. A pre-operative assessment of the tumor using 3D visualization techniques help in planning the surgical path and trajectory. The surgeon can then avoid the eloquent areas, blood vessels and other vital structures during surgery. This leads to excellent outcome for the patient with minimal complications and morbidity.

3.2 Related Work

In this section we discuss about the works already pursued for this task and their drawbacks which we have tried to overcome. We divide our discussion into three areas, namely, classical techniques, followed by deep learned techniques and finally mention those techniques that used both of the former techniques.

3.2.1 Classical techniques

Classical techniques such as level sets [105] and graph cuts [106] have been extensively used for segmentation [107], where, domain specific image features (e.g. gradient, intensity, texture, etc.) can be combined within an energy minimization framework. However, a majority of these algorithms rely heavily on initialization such as manual seeding, and are susceptible to segmentation errors due to unreliable location of the initial seed.

3.2.2 Deep learned techniques

In recent times, deep convolutional neural networks (CNN) have shown significant improvement, especially, in segmentation and classification problems [108–110]. However, traditional fully supervised techniques disregard domain knowledge which could help in better segmentation. CNNs for segmentation can be categorized based

on the dimension of convolutional kernel that is utilized. 2D CNNs employ 2D convolutional kernels for segmenting a single slice. Although, they are able to leverage contextual information across the height and width of the slice to make predictions, they are unable to extract any information from the adjacent slices. 3D CNNs mitigate this issue by using 3D convolutional kernels to predict 3D segmentation maps for a volumetric patch of a scan. This ability to grasp inter-slice context can improve the segmentation performance [111, 112].

However, 3D CNN comes with a computational cost as a result of the increased number of parameters used by them. Also, training 3D volumes at once consumes huge amount of computer memory making it difficult to train on large datasets. Recently, good segmentation performance have been achieved using a modified 3D CNN known as V-Net using Attention Guided (AG) , Squeeze and Excitation (SE) [113] and multi depth fusion modules [114].

3.2.3 Combination of classical and deep learned techniques

Combination of classical and deep learning techniques have been used in solving many medical imaging problems where data driven and domain specific approaches have been cascaded or ensembles of both techniques have been formed [115, 116]. We took inspiration from [115] where they mixed deep learning and graph cut to segment lung nodules. However, their work was restricted to 2D. They also did not modify the terms in the energy function of the graph cut, which can have significant impact on the segmentation performance. Furthermore, they used manual seeds for initialization of the graph cut algorithm. In sharp contrast, we have addressed the tumor segmentation problem in 3D. The data term, the smoothness term and the parameter which relatively weighs these two terms in the energy function of the graph cut are modified through a 3D deep learned model. We have further removed the need to manually initialize the seeds.

3.3 Contributions

In this work, we propose a deep graph cut model for segmenting brain tumors in 3D. The proposed approach combines data driven (graph cut) and domain specific (deep learning) strategies that are suitable for addressing the complexities of segmenting highly irregular structures like brain tumor. We term our solution Deep Graph Cut (DGC). DGC embeds deep learned probability maps of object (tumor) and background (everything other than tumor) voxels into the energy function of the graph cut. The deep learned voxel probabilities make the model highly robust to initialization errors. On the other hand, the unsupervised graph cut component can accurately segment structures through an energy minimization framework. To motivate the reader, we included a segmentation result using the proposed method in Fig. 3.1. In this figure we have shown two samples of MRI axial brain scans along with their corresponding segmented tumor. In Fig. 3.2, we also show how the segmentation looks in 3D both including and excluding the surrounding brain tissue region. We now summarize our contributions below:

1. Methodologically speaking, we propose a novel 3D segmentation framework by combining UNet and graph cut. In this framework, we derive new expressions for the data term, the smoothness term and the weighting parameter in the energy function of the graph cut from the probability maps of tumor and non-tumor voxels obtained from the UNet. Deep learned voxel probabilities from UNet make the model significantly robust to initialization errors. Unsupervised graph cut on the other hand offers flexibility to segment objects with intricate morphology.
2. From the application perspective, we solve 3D tumor segmentation, an important problem in brain imaging which has a far-reaching impact for the radiologists and the neurosurgeons.
3. We show a way to automate the evaluation of the manual parameter λ (used to balance the data term, and the smoothness term) in our Deep Graph Cut algorithm.

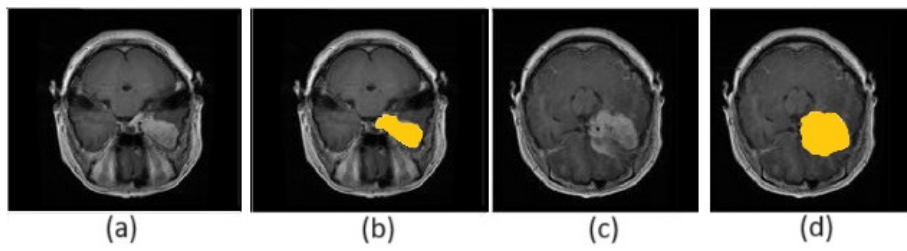


Figure 3.1: (a) A 2D axial slice view of an MRI brain scan, (b) segmented tumor region of (a), (c) 2D axial slice of another brain scan, (d) the corresponding segmented tumor output

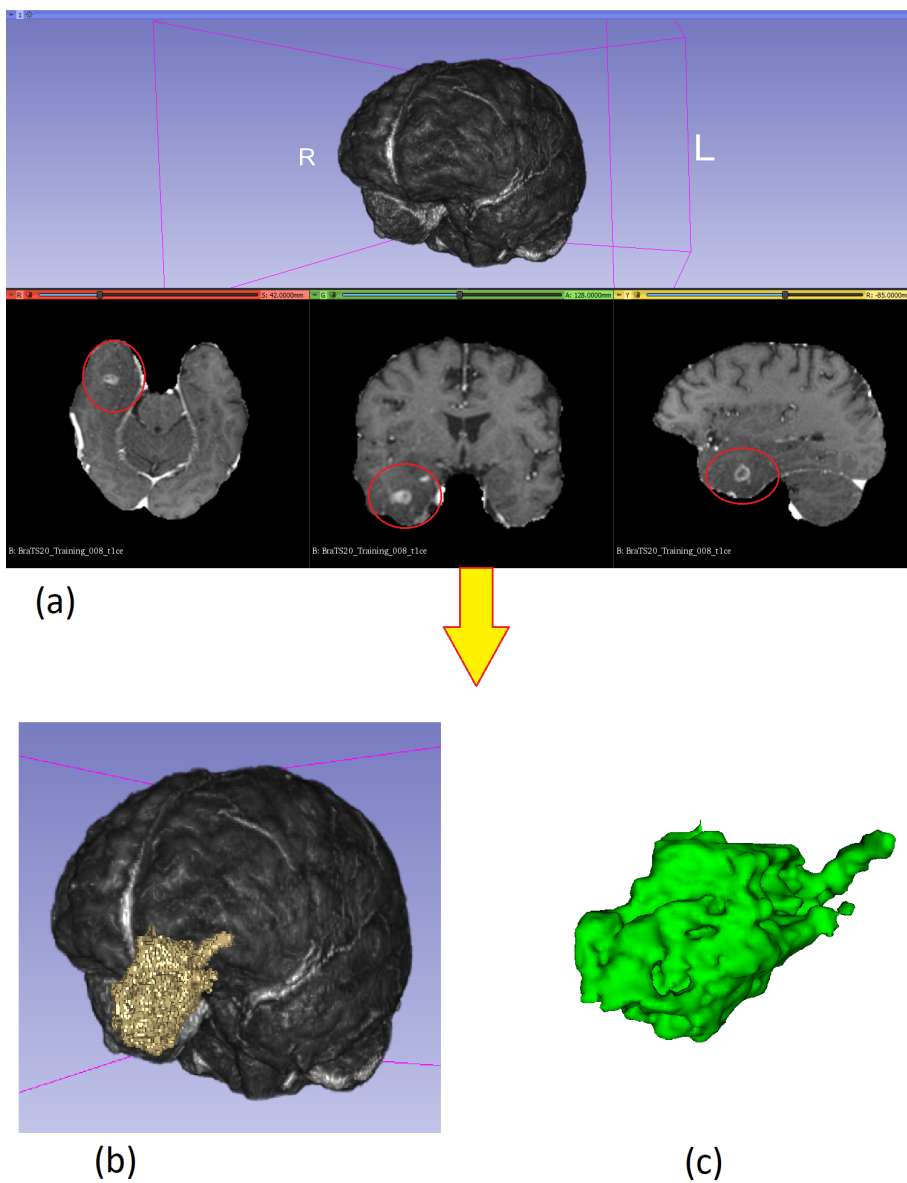


Figure 3.2: 3D view of the segmented tumor. (a) shows the input 3D MRI volume along with the axial, sagittal and coronal slice views. (b) shows the 3D segmented tumor along with the input brain scan (c) shows the 3D segmented tumor in isolation.

3.4 Proposed Method

In this section, 3D segmentation using UNet and graph cut are discussed first. We then describe in details our proposed solution (DGC). The section is completed with Algorithm 1 showing the steps of the DGC.

3.4.1 3D Brain Tumor Segmentation using UNet

We choose 3D UNet [33], a fully convolutional deep network architecture, as it is efficiently handles pixel-wise semantic labelling. UNet is an encoder-decoder network widely used for medical image segmentation [117]. This deep network can yield decent performance even in the absence of large amount of training data. The below Fig. 3.3 illustrates the U-Net model architecture. It comprises an expansive mechanism (right portion) and a contracting mechanism (left portion). Here the contracting mechanism is similar to the model of a convolutional neural network. This mechanism comprises two recurring $3 \times 3 \times 3$ un-padded convolutions. Each convolution is accompanied with a ReLU and a $2 \times 2 \times 2$ max pool operation using stride 2 for down-sampling. The count of the feature map is doubled at each of the down-sampling steps. The up-sample of the feature map at each step accompanied with a $2 \times 2 \times 2$ up-convolution as a part of the expansion mechanism. The up-sample reduces the count of feature channels by half. The feature map obtained from the expansion path is combined with the contraction block feature maps with the help of two operations, namely, one is convolution and another is ReLU operation. At each convolution, a crop operation is needed because of the reduced number of pixels at the border. Finally, a $1 \times 1 \times 1$ convolution is utilized with the output feature vector to determine the output class numbers. There are a total of 23 convolution layers in the overall network architecture. The up-sample operation in the architecture is designed such that the contextual information from the contracting path to expansive path can be maintained. This process finally makes the expansion block similar to the contraction block which generates a U-shape model. The network is trained for 500 epochs with stochastic gradient descent with initial learning rate of 0.0001, weight decay of 0.00001 and mini-batch size equal to 2 samples. We have used Adam

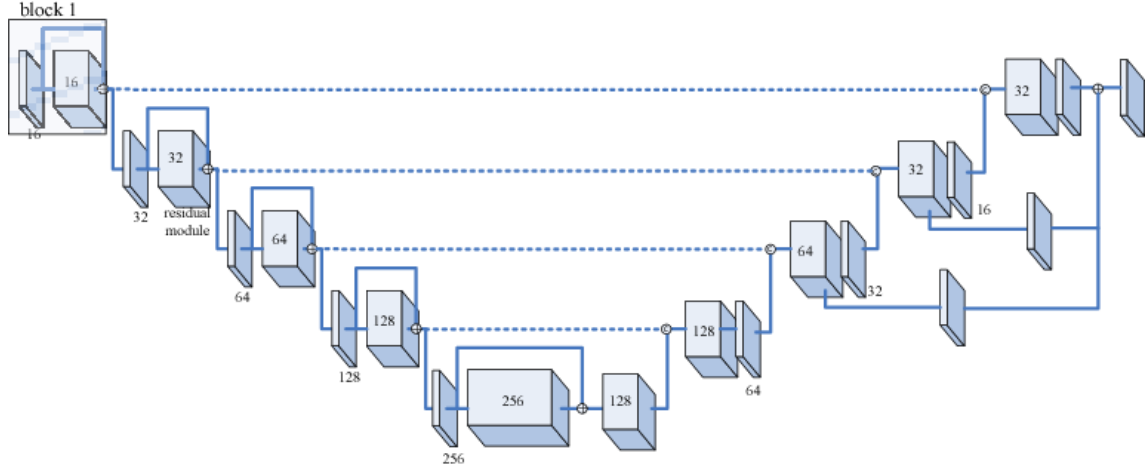


Figure 3.3: Architecture of 3D UNet. The numbers in each block represent the feature size

Optimizer and binary cross-entropy loss (as shown in Eq. 3.1) in the process. For 3D segmentation, the loss is computed over the entire volumetric data, where both the ground truth labels Y and the predicted segmentation \hat{Y} are 3D tensors of size (D, H, W) , where D is the number of slices (depth), H is the height, and W is the width of the volume. The binary cross-entropy loss is computed voxel-wise and then averaged over the entire 3D volume:

$$\mathcal{L}_C = -\frac{1}{N} \sum_i^{D \times H \times W} [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)] \quad (3.1)$$

where $N = D \times H \times W$ is the total number of voxels in the 3D volume. This ensures that the loss function is applied consistently across all voxels in the volume, capturing the full spatial information in 3D.

3.4.2 3D Brain Tumor Segmentation using Graph Cut

The input MRI is a 3D gray-scale volume which is represented by a 3D weighted graph $G = G(V, E)$. Each voxel x is a vertex in G and the set of all voxels is denoted by X . We consider two additional nodes, ‘source’ and ‘sink’, which are respectively denoted by s and t . We consider two types of edges/links, namely, the t-links (T) and the n-links (N). Each voxel x is connected to s and t via t-links. We use dense 26-neighborhood, denoted by $Ne(x)$ for each voxel x . Let y be a neighbor of x . So,

$y \in Ne(x)$ and we connect x and y via a n -link. Thus, $V = X \cup s \cup t$ and $E = T \cup N$. Let us define a segmentation A , which is a voxel-wise classification of all voxels into two classes, *i.e.*, either the "object" or the "background". Hence, following [106], we need to minimize the following energy function:

$$\zeta(A) = B(A) + \lambda R(A) \quad (3.2)$$

where $B(A)$ denotes the boundary properties/smoothness term and $R(A)$ denotes the regional properties/data term of A . Mathematically, these terms are expressed as:

$$B(A) = \sum_{x \in X, y \in Ne(x)} B_{(x,y)} \quad (3.3)$$

$$R(A) = \sum_{x \in X} R_x \quad (3.4)$$

where R_x and $B_{(x,y)}$ follow the standard definition of data and smoothness term in [118].

3.4.3 3D Brain Tumor Segmentation using Deep Graph Cut

The learned information from 3D UNet is embedded into the energy function of the 3D graphcut to achieve accurate segmentation. For each image, the 3D probability map is extracted from the last convolutional layer to calculate the probability of any voxel to belong to either tumor or background. In the proposed deep graph cut (DGC) model, we introduce a new data term, a new smoothness term and further combine the proposed data term and smoothness terms using a variable weight factor.

The 3D UNet estimates a regression function which maps each voxel of a 3D input image to a corresponding 3D voxel wise segmentation probability map $\mathcal{P} : \mathbb{R}^3 \rightarrow (0, 1)$ which also serves as an automated seed for the graph cut algorithm.

The deep learned data term is expressed as follows:

$$R_{DGC}(x)(Obj) = -\ln Pr(x = Obj)_{UN} \quad (3.5)$$

$$R_{DGC}(x)(Bkg) = -\ln Pr(x = Bkg)_{UN} \quad (3.6)$$

where the terms $Pr(x = Obj)_{UN}$ and $Pr(x = Bkg)_{UN}$ respectively indicate the probability of any voxel x to belong to the object (tumor) class and the background (anything other than tumor) class obtained from the UNet (UN). Naturally, $\forall x$, we have $Pr(x = Obj)_{UN} + Pr(x = Bkg)_{UN} = 1$. The negative log-likelihood assigns a negative value to the object pixels (voxels), and a positive score for background.

Brain MRI volumes tend to have low contrast between the tumor region and its surrounding structures thereby making the segmentation process difficult. In order to handle such constraints, we modify the smoothness term. The modification relies on the difference in the probabilities of any voxel and its neighboring voxel to belong to the tumor and the background classes. The improved smoothness term is given by:

$$B_{DGC}(x, y) = K_{(x,y)} \exp -\left(\frac{(I_x - I_y)^2}{2\sigma^2}\right) \times \frac{1}{d(x, y)} \times \frac{1}{\delta(x, y)_{DGC}} \quad (3.7)$$

where the term $d(x, y)$ denotes the Euclidean distance between two voxels x and y having intensity values I_x and I_y respectively. The term $K_{(x,y)}$ is based on the probabilities of these two voxels (x and y) to have the same segmentation class and is formulated as follows:

$$K_{(x,y)} = 1 - |Pr(x = Obj)_{UN} - Pr(y = Obj)_{UN}| \quad (3.8)$$

The value of $K_{(x,y)}$ is higher when the probability of two neighboring voxels to belong to the same segmentation class is similar, *i.e.*, when their absolute difference in probability values are less. The factor σ is chosen manually depending on image

noise level. The term $\delta(x, y)_{DGC}$ denotes the sum of differences between probabilities of neighboring voxels x and y to belong to the class tumor (object) and the class background. This can be expressed as:

$$\begin{aligned} \delta(x, y)_{DGC} = & |Pr(x = Obj)_{UN} - Pr(y = Obj)_{UN}| + \\ & |Pr(x = Bkg)_{UN} - Pr(y = Bkg)_{UN}| \end{aligned} \quad (3.9)$$

Note that the value of $\delta(x, y)_{DGC}$, for different regions (in fact for each voxel pair (x, y)) is calculated from the probability map derived from the output of UNet for the specific MRI input given. Hence, it is more robust and takes into account the variations in each input MRI encountered.

In the graph cut-based segmentation framework, the parameter λ regulates the relative importance of the smoothness term and the data term (see Eq. 3.2). It is crucial to properly choose this parameter as it largely influences the quality of the segmentation [106]. Ideally, the optimal value of λ should neither cause over-segmentation nor under-segmentation. However, finding an optimal value for this parameter remains a challenging problem. To remove the dependence on manual selection of this parameter, we propose in this work an automated calculation of λ , from the probability maps obtained from the 3D UNet. The calculation is based on the following careful observations:

1. If a voxel lies deep within the foreground or the background region, more importance should be given to the smoothness term. For such an interior voxel, the surrounding voxels generally have considerably higher probabilities to belong to the same segmentation class.
2. The data term is given more importance whenever a voxel lies nearer to the edge or boundary region. For such a terminal voxel, the surrounding voxels usually have much lower probabilities to belong to the same segmentation class.

Hence, we write $\lambda \propto \psi_{UN}(x)$, where $\psi_{UN}(x)$ is given by:

$$\psi_{UN}(x) = |Pr(x = Obj)_{UN} - Pr(x = Bkg)_{UN}| \quad (3.10)$$

Note that a higher $\psi_{UN}(x)$, which is determined from the probability maps in UNet denotes that the voxel is more towards the interior of the object (tumor). Unlike [119], where the authors used classical techniques to modify λ , we employed deep learned probability maps to modify it. Hence, the final form of λ becomes:

$$\lambda_{DGC}(x) = P\psi_{UN}(x) \quad (3.11)$$

where P is a constant which maps $\psi_{UN}(x)$ that lies between $[0, 1]$ to λ that lies between $[\lambda_{Min}, \lambda_{Max}]$. For our experiments, we have set $\lambda_{Min} = 1$ and $\lambda_{Max} = 100$ so that the data term has considerable impact on the energy function, which would not have been possible if λ_{Min} and λ_{Max} lies between $[0, 1]$. Finally, combining equations (3.5), (3.6), (3.7) and (3.11), we get the final modified energy function as shown below:

$$\begin{aligned} \zeta_{DGC}(A) = & \sum_{x \in X, y \in Ne(x)} B_{DGC}(x, y) + \\ & \lambda_{DGC}(x) \sum_{x \in X} R_{DGC}(x) \end{aligned} \quad (3.12)$$

Minimizing the above energy function via graph cut according to [106] will result in a 3D segmented image.

Algorithm 3.1: Deep Graph Cut (DGC)

Input: Trained 3D UNet model M , 3D graph G representing the 3D input MRI I , where V is the set of nodes/voxels and E is the set of edges

Output: 3D graph G_{out} with desired segmentation

- 1 $\mathcal{P} \leftarrow M(G)$
/* Applying M on G , we get the 3D probability map \mathcal{P} */
 - 2 Modify data term $R_{DGC}(x)$ for each voxel in $x \in G$ using \mathcal{P} following Equations 3.5 and 3.6
 - 3 Compute smoothness term $B_{DGC}(x, y)$ for each voxel x and its neighbor $y \in Ne(x)$ in G using \mathcal{P} as shown in Equation 3.7
 - 4 Compute weight factor λ for each voxel in $x \in G$ with \mathcal{P} using Equation 3.11
 - 5 Compute the final energy function using Equation 3.12, perform Graph cut and store the result in G_{out} .
 - 6 **return** G_{out}
-

3.5 Experimental Results

In this section, we first discuss data preparation. We then extensively evaluate our solution, including ablation studies and comparisons with external approaches. We implemented UNet in Tensorflow [120] and trained it on HP-Z640 workstation with 14 Core Intel Xeon processor, 128GB of memory and NVIDIA Titan RTX 24GB dedicated graphics processor.

3.5.1 Data Preparation

For experimentation, publicly available BRATS 2018 dataset [121, 122] is used. It is already split into training and validation sets containing 285 and 66 scans respectively. We use only T1-weighted contrast-enhanced (T1CE) sequences from each scan for our experiments, as we found T1CE best represents the data in terms of contrast. The data were publicly made available with some pre-processing, *i.e.* they were co-registered to the same anatomical template, skull-stripped and interpolated to a resolution of $1mm^3$. As all the scans were stored in compressed Nifti image format (*i.e.*, .nii.gz), we used SimpleITK library [123] to convert nifti format images into 3D numpy array [124]. We further normalized all input images to have zero mean and unit standard deviation (std) based on non-zero voxels only. For augmentation, we applied a random intensity shift ($-0.1 \dots 0.1$ of image intensity std) and scale ($0.9 \dots 1.1$) on the input images. We also applied random horizontal flips (for x-axis) with a probability 0.5. As UNet requires the input image with a specific resolution [125], and to reduce the memory constraints while minimizing information loss, we crop each image volume from its original resolution ($240 \times 240 \times 155$) to a resolution of ($192 \times 192 \times 144$). As the λ parameter is learned from the UNet, only one parameter (σ) that is used in equation (3.7), is manually set to 4 during the training phase after various trials.

3.5.2 Performance Metrics

For the quantitative performance evaluation, Dice Similarity Coefficient (DSC) is employed [126]. DSC is a statistical metric used to measure the *overlap* between

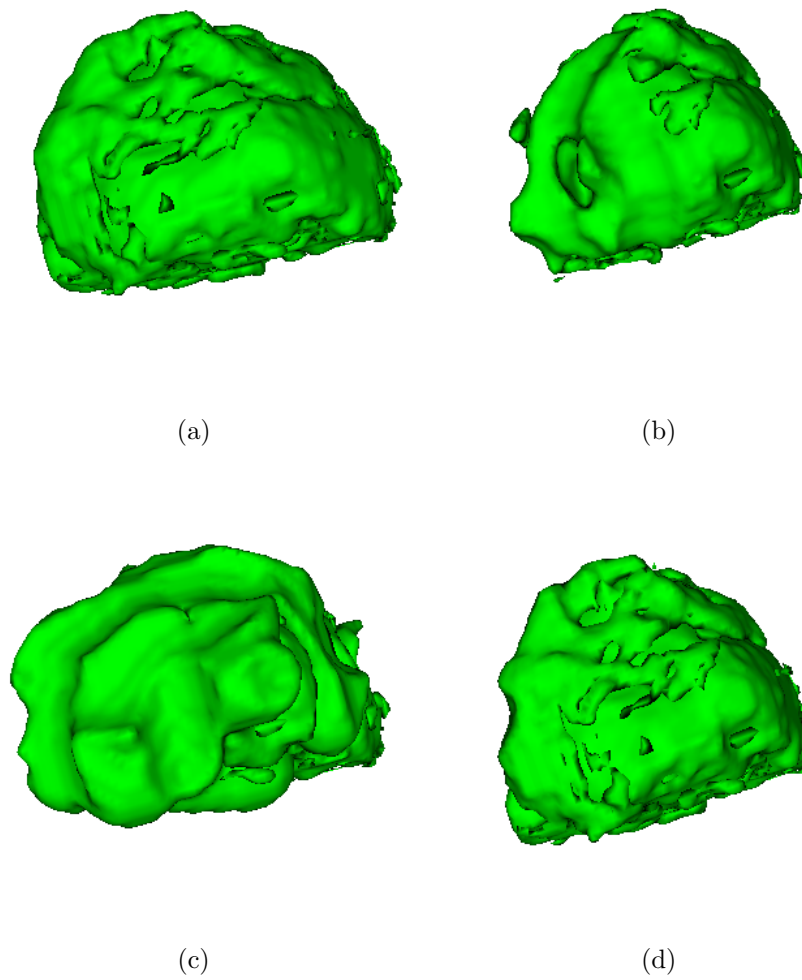


Figure 3.4: Qualitative comparisons for one dataset: (a) Ground Truth (GT), (b) Segmentation using 3D UNet (DSC = 0.862), (c) Segmentation using Graph Cut (DSC = 0.812), (d) Segmentation using DGC (DSC = 0.91)

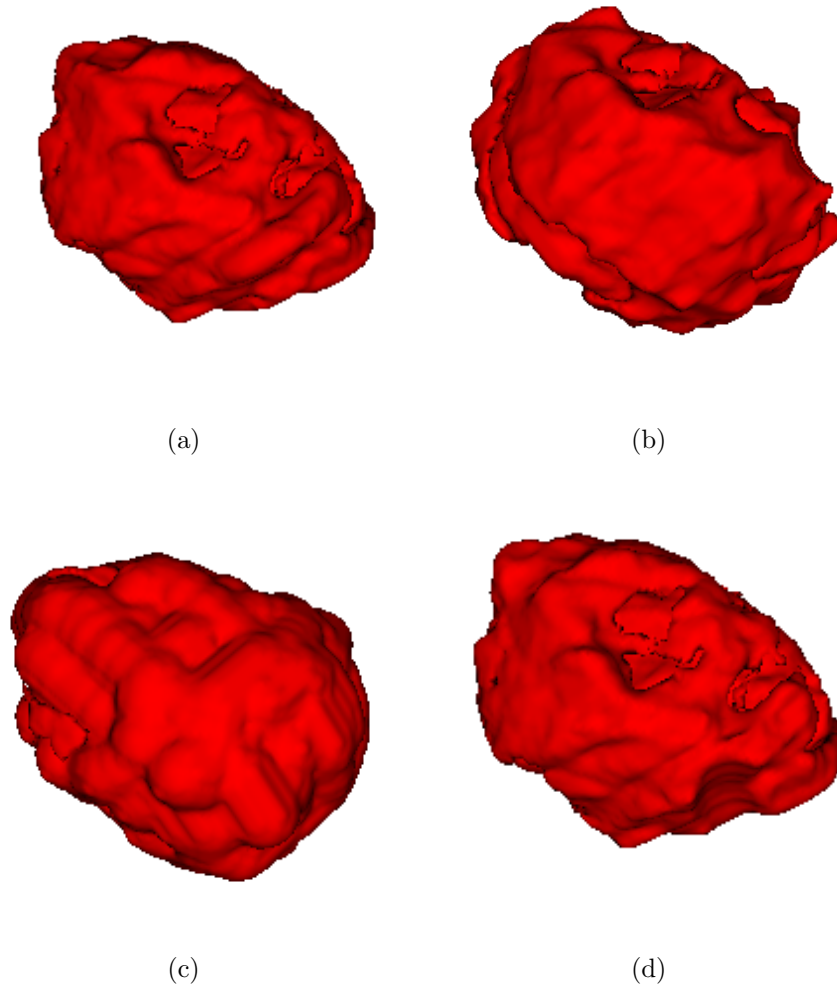


Figure 3.5: Qualitative comparisons for a second dataset: (a) Ground Truth (GT), (b) Segmentation using 3D UNet (DSC = 0.877), (c) Segmentation using Graph Cut (DSC = 0.897), (d) Segmentation using DGC (DSC = 0.934).

two sets, commonly applied in segmentation tasks to quantify the agreement between the predicted segmentation and the ground truth. It is defined as:

$$DSC = \frac{2|J \cap K|}{|J| + |K|} \quad (3.13)$$

where:

- J is the set of voxels (or pixels) in the predicted segmentation.
- K is the set of voxels in the ground truth segmentation.
- $|J \cap K|$ represents the number of overlapping voxels (true positives).
- $|J|$ and $|K|$ are the total number of voxels in the predicted and ground truth segmentation, respectively.

DSC ranges from 0 to 1, where:

- $DSC = 1$ indicates a perfect match between the predicted segmentation and the ground truth.
- $DSC = 0$ means no overlap at all.

Comparison Between Ground Truth and Segmented Output in 3D

We compute the Dice coefficient over the entire 3D volume, rather than on individual 2D slices. The equation for DSC in voxel-wise comparison across a 3D volume is:

$$DSC = \frac{2 \sum_{i=1}^N y_i \hat{y}_i}{\sum_{i=1}^N y_i + \sum_{i=1}^N \hat{y}_i} \quad (3.14)$$

where:

- $N = D \times H \times W$ is the total number of voxels in the 3D volume.
- y_i represents the binary ground truth label for voxel i ($y_i = 1$ for foreground, $y_i = 0$ for background).
- \hat{y}_i is the predicted label for voxel i ($\hat{y}_i = 1$ for foreground, $\hat{y}_i = 0$ for background).

Table 3.1: Ablation study I: Graph cut, Graph Cut enhanced with combinations of (a) DGC data term (b) DGC smoothness term (c) DGC variable weight factor. Best values are shown in **bold**.

Modification	DSC
Graph Cut (Boykov <i>et al.</i> [106])	0.53 ± 0.17
GC with (a) and (b) only	0.883 ± 0.23
GC with (b) and (c) only	0.856 ± 0.19
GC with (a) and (c) only	0.87 ± 0.034
DGC (GC with (a), (b) and (c))	0.92 ± 0.073

Table 3.2: Ablation study II: Graph Cut, 3D UNet, DL-GC in 2D and DGC. Best values are shown in **bold**.

Algorithm	DSC
Graphcut (Boykov <i>et al.</i> [106])	0.53 ± 0.17
3D UNet (Çiçek <i>et al.</i> [33])	0.83 ± 0.034
DL-GC in 2D (Mukherjee <i>et al.</i> [115])	0.82 ± 0.062
DGC	0.92 ± 0.073

3.5.3 Ablation Studies

We have conducted two ablation studies. The first ablation study is undertaken to demonstrate the individual contributions of the data term, smoothness term and the variable weight factor in the proposed DGC algorithm. The results are shown in Table 3.1. For all the combinations, UNet was used. As is evident from this table, all the three terms contribute towards improving the performance of the DGC algorithm. We also conduct a second ablation study to examine the contributions of the 3D UNet and the Graph Cut used in isolation. Qualitative segmentation outputs for two datasets are illustrated in Fig. 3.4 and 3.5. We also implement Mukherjee *et al.*'s work [115] where they have modified only the data term of Graph cut using 2D UNet and required manual seeding, but we automatically initialized the seed for Graph Cut using [127]. Table 3.2 indicates that DGC algorithm clearly outperforms basic 3D UNet [33], vanilla Graph cut [106] and 2D-DLGC algorithm [115].

Table 3.3: Comparison of State of the art Methods. Best values are shown in **bold**.

Algorithm	DSC	Sensitivity	Specificity
EM Graphcut [128]	0.70 \pm 0.21	-	-
Random Walks [129]	0.72 \pm 0.08	-	-
Anfis Graphcut [130]	0.85	-	-
3D CNN [131]	0.85	0.877	-
SegNet 2D [132]	0.85	-	-
AGSE V-Net [113]	0.85	0.83	0.99
No New-Net [133]	0.878	-	-
2D CNN [134]	0.88	0.84	0.88
NVDLMED [135]	0.8839	-	-
2D M-UNet [136]	0.90	-	-
UNet Ensemble [137]	0.90	-	-
DGC	0.92 \pm 0.073	0.94 \pm 0.003	0.91 \pm 0.008

3.5.4 Comparison with State-of-the-art Methods

We now compare our method with other state-of-the-art segmentation approaches in table 3.3. The methods are diverse in nature, with some of them using deep learning and others are not. Among the methods which have not employed deep learning, we choose two Graph Cut based methods, one on random walks [129], one method using Competitive Expectation Maximization Algorithm [128] and another using normalization techniques [130]. Among the deep learning based methods, We use a 2D CNN [134], a 3D CNN [131], V-Nets [113], BRATS 2018 winning solutions [133, 135] and an ensemble 3D UNet [137]. The results of comparisons in Table 3.3 are shown only for whole tumor segmentation with three metrics namely Dice Score, sensitivity and specificity, and it clearly establishes the superiority of our approach over all the competing methods. Note that most of the published works did not report their sensitivity and specificity values.

3.5.5 Execution Time

During the testing phase, we have used a PC with Intel(R) Core (TM) i7 9th Gen CPU processor having a base clock speed of 2.6 GHz and 16 GB RAM. Average time for segmenting a 3D MRI dataset with resolution of $192 \times 192 \times 144$ was 278s.

3.5.6 Time Complexity Analysis

We have used 3D UNet and graph cut for segmentation of brain tumor in this work. The time complexity of 3D UNet and graph cut is explained in Sec. 2.3.3 (Eq. 2.25) and 10 respectively. Combining both the time complexities, we can represent the overall time complexity for our solution as-

$$\begin{aligned} T_{DGC} &= T_{UNet} + T_{GC} \\ &= O\left(\sum_{n=1}^{4L} k_{n-1} \cdot s_n^3 \cdot f_n \cdot l_n^3 \cdot r_1 \cdot b_1 + V \cdot E^2\right) \end{aligned} \quad (3.15)$$

where d is the number of convolutional layers, l_n is the length of the output feature map in the n th layer, f_n is the number of filters in the n th layer, s_n is the filter size in the n th layer, k_{n-1} is the number of input channels, r_1 is the learning rate, b_1 is the batch size, V is the number of vertices/nodes in the graph and E is the number of edges in the same graph.

3.6 Compliance with ethical standards

This research study was conducted retrospectively using human subject data made available in open access by [121,122]. Ethical approval was not required, as confirmed by the license attached with the open access data.

3.7 Summary

Detecting, delineating, and characterizing 3D brain tumors using MR imaging is crucial for treatment planning. This work demonstrates how combining UNet with graph cut improves 3D segmentation performance. We derive new expressions for the graph cut energy function using probability maps from UNet. Our deep graph cut model shows competitive performance on the BRATS dataset through extensive testing.

Chapter 4

Brain Tumor Classification

The development of brain tumor classification techniques using multimodal data, such as radiological and histopathological inputs, is crucial for improving diagnostic precision and tailoring treatments in clinical practice. By integrating diverse data sources, these techniques provide a comprehensive understanding of tumor characteristics, leading to more accurate classifications and personalized therapies, which enhance patient outcomes. Ultimately, the goal is to streamline the diagnostic process, reduce uncertainty, and support more informed decision-making, thereby leading to more effective and timely interventions in real-world clinical settings.

4.1 Introduction

As the most common primary malignant tumor of the central nervous system, glioma comprise approximately 100,000 newly diagnosed cases each year [138]. Due to the complex tumor micro-environment and spatial heterogeneity, Magnetic Resonance Imaging (MRI) itself is insufficient for complete characterization of gliomas (e.g. grading and sub-typing). Thus, histopathology examination is often required [139]. In histopathology, gliomas are classified based on the morphological features of the glial cells including increased cellularity, vascular proliferation, necrosis, and infiltration into normal brain parenchyma [140, 141]. Histopathological classification of brain tumours is dependent on the recognition of areas with the characteristic histopathology for a particular tumour type. Classification of gliomas into astro-

cytoma, oligodendroglioma and glioblastoma is the foundation upon which relies the prognosis, treatment and management of the patient. Microscopic examination is the gold standard, however there is great inter-observer variability based on the subjective evaluation. Use of machine learning algorithms utilising the particular histopathological characteristics of individual tumours, independent of subjective analysis can help reduce this diagnostic variability. With advances in digital pathology and whole slide imaging, morphology based automated pathologic diagnosis has become a reality.

4.2 Related Works

The 2018 instance of the Computational Precision Medicine Radiology-pathology challenge [142] is the first effort towards providing a publicly available dataset with pairs of Whole Slide Images (WSI) and MRIs. Sahayam *et al.* [143] discarded histopathology data and used only radiology data to perform classification and that resulted in lower classification accuracy. Chan *et al.* [144] grouped WSI tiles into several clusters in an unsupervised manner and applied a random forest for final prediction. Xue *et al.* and Pei *et al.* [145, 146] both used tumor segmentation followed by tumor classification on MRI volumes but performed a three-way simultaneous classification that yields lower accuracy than binary classification. Hamidinekoo *et al.* [147] also used three-way simultaneous classification and used 2D CNN for both radiology and histopathology data. 2D CNN used for slice wise classification in 3D MRI volumes is inefficient as 3D CNN captures more spatial information. The top-performing method [148] used a soft-voting ensemble based on a radiographic model and a histologic model. The MRI model relies on radiomics with prior automatic tumor delineation, while the histologic model classifies patches extracted from WSIs which are filtered using an outlier detection technique. The second-best performing method [149] also used two models for both modalities. The radiographic model is an end-to-end deep learning approach, while the histologic model uses a deep learning model to perform feature extraction by dropping its last layer, to further classify a set of extracted tiles into a WSI prediction. Ma *et al.* [150] used two convolutional

neural networks for radiology and pathology images respectively. ResNet34 and ResNet50 were directly applied to extract features from WSI grayscale patches and classified them. 3D DenseNet was employed for MRI sequence. A further regression model was introduced for the inference. Pei *et al.* [151] segmented the brain tumor from the MRI sequence, and then classified it by a regular 3D CNN model. But they did not discover the massive information in WSI.

4.3 Contributions

In this chapter, we propose an effective pipeline for multi-modal tumor classification among glioblastoma, astrocytoma and oligodendroglioma using radiological and histopathological images. Our main contributions are as follows-

1. We apply a 3D CNN model for a coarse classification, *i.e.* glioblastoma vs. non-glioblastoma (which could be either astrocytoma or oligodendroglioma) from 3D MRI volumes.
2. We construct a deep feature extraction model for WSI using 2D CNN. Features from two different magnification levels of the WSI are treated as local and global features.
3. A feature vector combining the local and global features is used as nodes/vertices for a Graph Convolutional Network (GCN). This GCN with normalized graph Laplacian, used to perform the classification of non-glioblastoma into astrocytoma and oligodendroglioma.

4.4 Proposed Method

Our proposed method primarily consists of three major steps, namely, i) radiological data classification, ii) histopathological feature extraction and iii) classification of histopathological data using GCN. We first perform a coarse classification from the radiological data by inferring presence or absence of Glioblastoma. This is possible as Glioblastoma exhibit distinctive macro features which can be identified from the

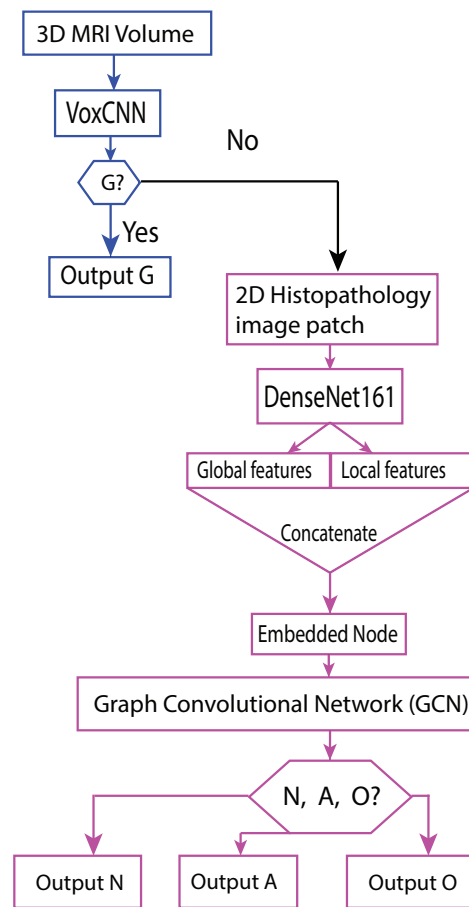


Figure 4.1: Flowchart of our classification model: 'G' stands for Glioblastoma, 'N' stands for Normal, 'A' stands for Astrocytoma and 'O' stands for Oligodendroglioma.

MRI data. However, 3D MRI volumes provide very little information about tumor subtypes like oligodendroglioma and astrocytoma. So, we use histopathological data for that purpose. The overall flowchart of our classification technique is portrayed in Fig.4.1.

4.4.1 Radiological phase

In this step, we consider only radiological data, *i.e.*, 3D MRI volumes. Each subject has 4 different types of MRI, namely, T1, T2, T1CE, and flair. We perform binary classification by dividing the data into two groups for each MRI type, one group consists of only glioblastoma subjects and the other group consists of astrocytoma and oligodendroglioma subjects.

(A) Data pre-processing and augmentation

Each MRI volume is first corrected using the N4 bias field correction algorithm [152], a popular method for correcting low frequency intensity non-uniformity present in the MRI image data. The original resolution of each corrected volume was $240 \times 240 \times 155$. Due to memory constraints, this volume was down-sampled to a resolution of $128 \times 128 \times 128$ using cubic interpolation re-sampling algorithm from SimpleITK library [153].

Since, the number of subjects belonging to each class is different we apply horizontal flipping along the sagittal plane on the volumes belonging to minority class to balance the samples of each class.

(C) Classification model

The classification was based on VoxCNN [154] with appropriate modifications. In this case, VoxCNN is trained from scratch for each of the four MRI types separately. VoxCNN architecture has four volumetric convolutional blocks for extracting features (with a number of filters increasing from layer to layer), two deconvolutional layers with batch normalization and dropout for regularization and an output with SoftMax nonlinearity for classification. We have kept the network architecture as it is defined in the article [155]. The input files are in nifti format, and they are normalized between 0 and 1 for keeping similarity among all models. Considering the dataset size, model size and the limitations of GPU memory, we modified the batch iteration process in order to get samples of each class in every batch. The reason being that the probability of having only one class represented inside a batch for infinite number of samples is $\frac{1}{c^b}$ where c is the number of classes and b is the batch size. Therefore, for large batch sizes this probability is low. However, for our problem, it is high enough to thwart the learning process. Balancing of the samples inside each batch was hence undertaken to obtain stable learning curves. Modulated Rank Averaging (MRA) method is applied next to perform weighted voting among the 4 MRI types to arrive at the final prediction. In the traditional majority voting method, the final prediction is based on the majority of classifiers'

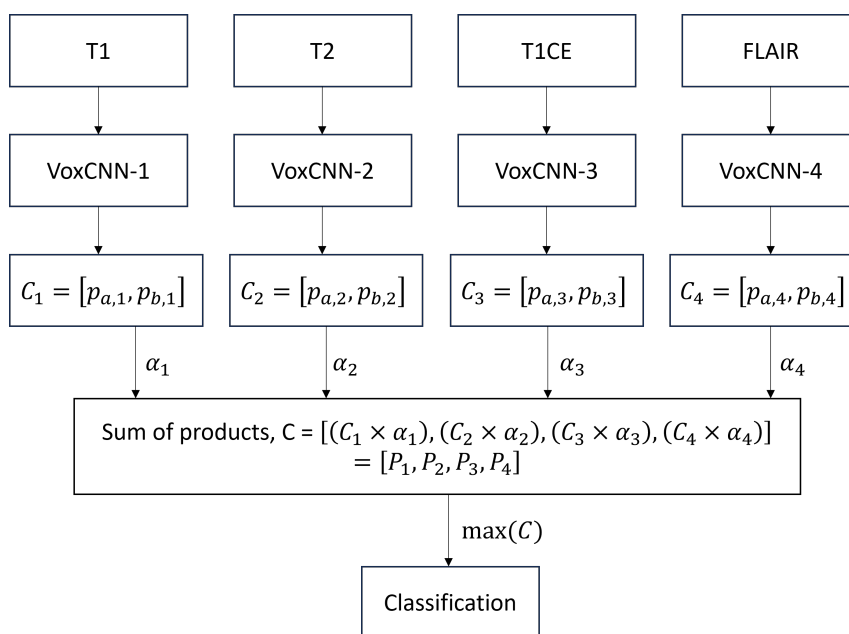


Figure 4.2: Overall workflow for radiological classification phase

outputs, but for multi-class tasks, it can be ineffective. While single classifiers may perform well, difficult cases can result in higher error rates due to uncertainty among categories. Our approach uses weight-adjusted probability vector fusion and ranks classifiers based on their individual accuracy. Classifiers with lower training error receive greater weight when combining results during testing. We developed the MRA technique in [154] and is described in details in Chapter 6. The overall workflow for the radiological classification is shown in Fig. 4.2 where a, b are the labels that need to be classified which is Glioblastoma and non-glioblastoma in our case. We have numbered each VoxCNN model from 1 to 4 corresponding to the four types of 3D MRI inputs - T1, T2, T1CE and FLAIR. C_1 is the predicted probability vector of model 1, $p_{a,1}$ means the predicted probability by model 1 for the image to be in class a . Other symbols have similar meanings. We finally do a softmax classification in the end denoted as $\max(C)$.

4.4.2 Histopathological phase

The process of selecting representative patches to be used during the training phase is very important and can be quite time-consuming. Weakly-supervised approaches [156] only use the slide labels during the training of the aggregation model. In

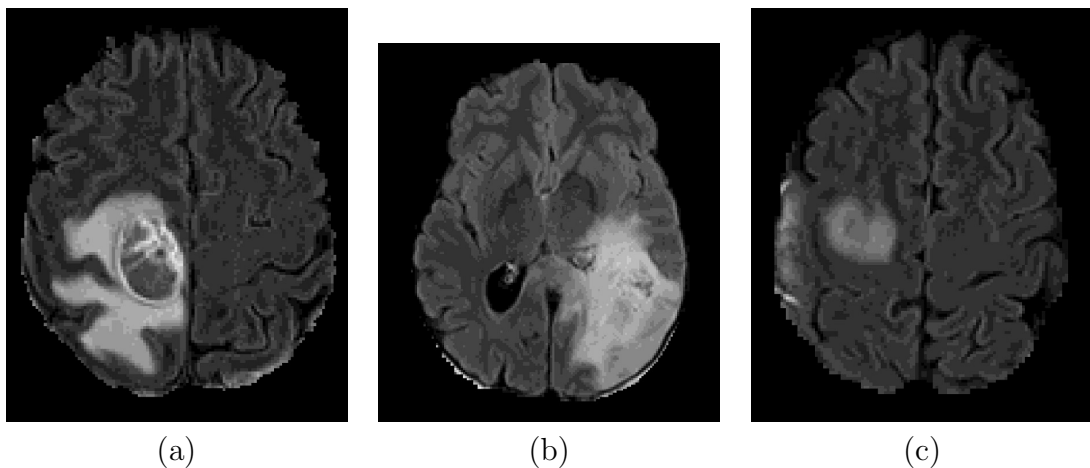


Figure 4.3: Axial flair MRI images of (a) Glioblastoma Multiforme, (b) Astrocytoma, (c) Oligodendroglioma. It can be seen in case of (a) the tumor region has thick demarcations with a central dark necrotic region and an irregular whitish edema region while in cases of (b) and (c) there is no such distinctive feature

contrast, we adopted a semi-automated approach for optimal selection of tiles.

(A) Patch selection

To select the most important tiles to classify WSIs, we have adopted a two-stage approach.

Stage 1: We extract tiles/patches of size 1000×1000 pixels without any overlaps from each WSI using deep zoom extractor provided in the openslide framework [157]. Each WSI produces patches from different magnification levels. We select two such levels for our work. The first one has the highest level of magnification (20x) representing the local features and the other is 3 times zoomed out (17x), which represents the global features. After experimentation, we found selecting two such levels give the best representation of each WSI.

Stage 2: The histopathology images contain patches that include air bubbles, cracks, and blurred regions. We remove those tiles by studying the histogram of the images. For this purpose, we devised an automated technique selecting only those images that have a balanced histogram, *i.e.*, those images that have higher number of pixels between intensity range 0 - 192 than those in the range 192 - 255 in all three color channels. After careful observations, we found that the artifacts mentioned above are mostly grayscale and hence have most of the pixels with inten-

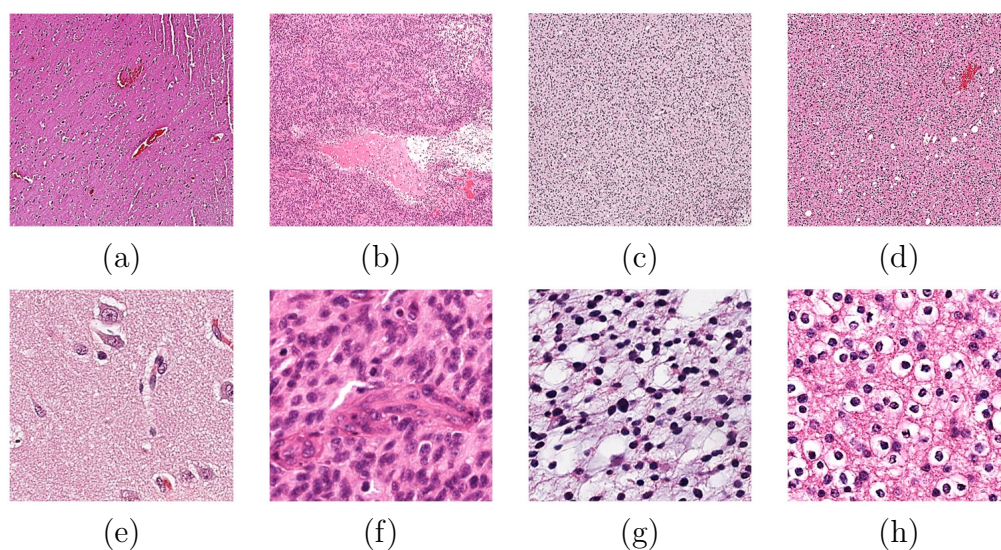


Figure 4.4: Histopathology images of (a) normal, (b) glioblastoma, (c) astrocytoma and (d) oligodendroglioma cells with global magnification level. Corresponding images (e), (f), (g) & (h) are shown with local magnification level.

sity greater than 192 in all three channels. Here, the lowest intensity 0 represents white and highest intensity 255 represents black.

Stage 3: Even after removing unnecessary patches, some of them do not contain the necessary cellular information, as the tumor has not spread throughout the whole tissue scanned in the laboratory. Some regions had healthy cells, while others had tumor cells distinctly visible. Our neuro-pathologists manually select patches that contain only healthy cells labelled as 'N' (normal) and patches that contain more than 80% tumor cells, labelled as 'A' (astrocytoma) or 'O' (oligodendroglioma). Example images of normal, glioblastoma, astrocytoma and oligodendroglioma at two different magnification levels are shown in Fig. 4.4. We have selected equal number of patches for each class from each magnification level to prevent class imbalance. The extracted tiles were then randomly investigated visually to reassure false tiles were not included in the training set.

Deep feature extraction

We apply DenseNet model [158] with 161 layers to extract the deep features from each patch. We chose DenseNet over other popular neural networks like VGGNet or ResNet because DenseNet is more complex than VGGNet and achieves better results

using fewer parameters than ResNet. DenseNet is more complex than VGGNet due to its dense connectivity, where each layer receives inputs from all preceding layers, leading to better feature reuse and gradient flow. Unlike VGGNet, which consists of sequential convolutional layers, DenseNet reduces redundancy by explicitly reusing features, making it more parameter-efficient. Compared to ResNet, which uses residual connections to ease gradient propagation, DenseNet achieves better results with fewer parameters by minimizing redundant feature learning and encouraging feature reuse across layers. The last two layers of DenseNet are modified to extract features of 1000 dimensions. The model is trained to classify among three classes ('A', 'O' and 'N'). Let the feature vector F_i for an image i ($i = 1, \dots, M$) be represented as:

$$F_i = [F_i^{g(1)}, \dots, F_i^{g(1000)}, F_i^{l(1)}, \dots, F_i^{l(1000)}] \quad (4.1)$$

$$F_i^{g(j)} = \frac{\sum_{x=1}^S F_x^{g(j)}}{S}, j = 1, \dots, 1000; x \in i \quad (4.2)$$

and

$$F_i^{l(j)} = \frac{\sum_{x=1}^T F_x^{l(j)}}{T}, j = 1, \dots, 1000; x \in i \quad (4.3)$$

Here, $F_i = [F_i^{(j)}, j = 1, \dots, d_f]$ ($d_f = 2000$) is obtained by concatenating 1000 dimensional global feature vector $F_i^g = [F_i^{g(j)}, j = 1, \dots, 1000]$ and the 1000 dimensional local feature vector $F_i^l = [F_i^{l(j)}, j = 1, \dots, 1000]$. The total number of patches (x denotes a patch) at the global level is denoted by S and that at the local level is denoted by T . The global and local features are differentiated by their magnification levels. The local features are highly magnified (at 20X Zoom level) whereas the global features are less magnified (at 17x Zoom level). For this work, $S = 150$ and $T = 1000$. For each component of the global and local feature vector, averaging is done over all the patches.

4.4.3 Graph Convolutional Network

To make more precise and accurate decisions on classification, we employ graph convolutional network (GCN) which can effectively capture the relation-aware representation (RAR). Most of the classification models provide attractive results when provided with a huge set of labeled samples. However, GCN can perform well with fewer training samples, which is the case for the present problem. RAR can provide the necessary information to guide the training with fewer examples. GCN can help generalize the standard convolution operation to graph convolution [34].

(A) Graph Construction

We construct a graph $G = G(V, E)$, with M nodes $v_i \in V, i = 1, \dots, M$ and edges $e_{ij} = (v_i, v_j) \in E$. Here, each node v_i is represented by the concatenated feature vector F_i and M denotes the number of image samples in the training set.

(B) Construction of Adjacency Matrix

The adjacency matrix, denoted as A , defines the relationships (or edges) between vertices. Each element in A can be generally computed by using the following radial basis function (RBF).

$$A_{i,j} = \begin{cases} 1 & \text{if } \exp\left(-\frac{\|v_i - v_j\|^2}{\sigma^2}\right) > \lambda \\ 0 & \text{otherwise} \end{cases} \quad (4.4)$$

where, σ is a parameter to control the width of the RBF kernel, and λ is the similarity threshold, both of which are chosen manually. λ determines the threshold above which a pair of nodes should be connected. It is used to prevent the construction of a highly connected (or even complete) graph, which will be computationally expensive and redundant. We next obtain the graph Laplacian matrix L as follows:

$$L = D - A \quad (4.5)$$

where D is a diagonal matrix representing the degrees of A , *i.e.*, $D_{i,i} = \sum_j A_{i,j}$. GCNs make use of the eigen-decomposition of graph Laplacian matrix to implement information propagation within graph [34]. To enhance the generalization ability of the graph, the symmetric normalized Laplacian matrix (L_{sym}) can be used as follows:

$$\begin{aligned} L_{sym} &= D^{-\frac{1}{2}} L D^{-\frac{1}{2}} \\ &= I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \end{aligned} \quad (4.6)$$

where I is the identity matrix. Thus, GCN is able to apprehend the RAR feature [159].

(C) Graph Convolutional Layers

The point of GCN is to express G through a neural network model $f(X, A)$ in which $X \in R^{N \times d_f}$. Mathematically, a multi-layer GCN updates all feature representation for all nodes via the layer-wise rule:

$$H^{(l+1)} = \sigma(L_{sym} H^{(l)} W^{(l)}) \quad (4.7)$$

where σ is the ReLU function. $H^{(l)} \in R^{N \times d_l}$ stands for the feature representation of l^{th} layer. Considering $X = H^{(0)}$, a two-layer GCN (2L-GCN) has the following layers:

$$H^{(1)} = \sigma(L_{sym} \times W^{(0)}) \quad (4.8)$$

$$H^{(2)} = \sigma(L_{sym} H^{(1)} W^{(1)}) \quad (4.9)$$

where $W^{(0)} \in R^{d_0 \times d_1}$, and $W^{(1)} \in R^{d_1 \times d_2}$ are two trainable weight matrices. The node feature set V and adjacency matrix A were passed to a 2L-GCN, and we obtained $H^{(2)} \in R^{N \times D}$ where $D = d_f$. The $H^{(2)}$ was then combined with I via dot

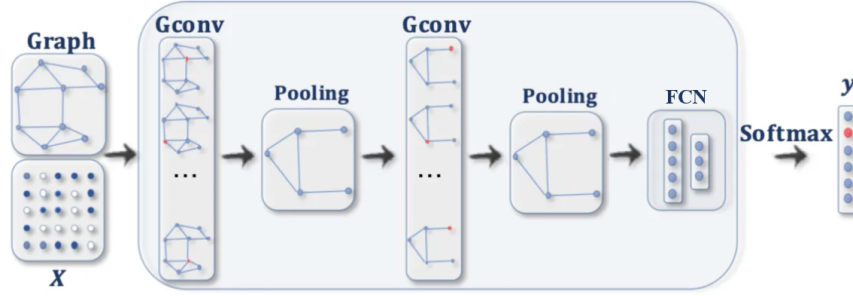


Figure 4.5: A coarse schematic of a two layer GCN with a fully connected neural network classifier at the end. Source: [8]

product-

$$y = H^{(2)}I \quad (4.10)$$

Through a linear projection (LP) with trainable weights $W^{(2)} \in R^{N \times N_C}$, in which N_C denotes the number of classes,

$$z = yW^{(2)} + W^b \quad (4.11)$$

where $z \in R^{N_C}$, and W^b represents the bias. $N_C = 3$ in our case. Hence, we only need to train $(W^{(0)}, W^{(1)}, W^{(2)})$ and cognate biases for this 2L-GCN. A coarse schematic of a 2L-GCN is shown in Fig. 4.5, where X is the input adjacency matrix and y is the output class label. Here, FCN denotes a fully connected neural network which acts as a classifier and classifies the given features to produce the output y .

4.5 Experimental Results

In this section, we first describe the dataset. This is followed by a discussion on the model hyper-parameters, and various other parameters used. Then, we present an ablation study. Finally, we show the comparisions with several atate-of-the-art approaches.

4.5.1 Dataset

To train and validate our pipeline, we used the CPM-RadPath 2020 challenge dataset [160]. The dataset consists of multi-institutional paired MRI scans and WSIs of brain gliomas, obtained from the same patients. Each subject, belongs to one of the following class- 'A' (astrocytoma), 'O' (Oligodendroglioma) and 'G' (Glioblastoma). There are three data subsets: training, validation, and testing with 270, 78, and 40 co-registered radiology and histopathology subjects respectively. The radiology data consists of four modalities: T1, T2, T1CE, and flair. The data after pre-processing, co-registered to the same anatomical template, interpolated to the same resolution ($1mm^3$), and skull-stripped is found to be distributed. The histopathology data contains one WSI for each patient, captured from H&E stained tissue specimens.

4.5.2 Training VoxCNN, DenseNet and GCN

All the computations are done in HP Z640 Workstation with Intel Xeon 14-core Processor having 128GB Random Access Memory (RAM) and NVIDIA Titan RTX 24GB Graphics processor using PyTorch 1.9 [161] and PyTorch Geometric [162] in Ubuntu 20.04. For the radiological classification (Section 4.4.1), the batch size, learning rate and total number of epochs are 16, 27×10^{-6} and 200 respectively. For the histopathology image classification (Section 4.4.2), each image was resized to 224×224 as required by the DenseNet model. The batch size, learning rate and total number of epochs are 32, 3×10^{-7} and 300 respectively. For the construction of the adjacency matrix required to train GCN (Section 4.4.3), the σ and λ parameters are experimentally set to 5 and 0.75 respectively. The GCN is trained for 1200 epochs with a learning rate of 3×10^{-4} . Adam optimizer and cross entropy loss function are used for 3D CNN, 2D CNN and GCN.

4.5.3 Ablation Study

An ablation study is undertaken to demonstrate the impacts of individual modalities, *i.e.*, radiology and histopathology. Table 4.1 presents the results of the ablation study, which evaluates three different scenarios:

Table 4.1: Ablation study of various combinations of modality

Approach	F1 Score	Cohen’s Kappa	Balanced Accuracy
MRI Only	0.668	0.632	0.76
WSI Only	0.812	0.787	0.86
MRI+WSI	0.928	0.892	0.904

1. Radiological Features Only: Classification of tumor types based solely on radiological features.
2. Histopathological Features Only: Classification of tumor types based solely on histopathological features.
3. Combined Features: Classification using both radiological and histopathological features.

The proposed pipeline is designed specifically for the third scenario, where both types of features are utilized. To evaluate the first scenario, an alternative model, VoxCNN, is trained to classify all tumor types instead of the original VoxCNN model, which was trained to distinguish between glioblastoma and non-glioblastoma. Similarly, for the second scenario, an alternative DenseNet model is trained in conjunction with a Graph Convolutional Network (GCN) to classify all tumor types, replacing the DenseNet+GCN model used in the proposed pipeline.

These experiments demonstrate that relying solely on radiological or histopathological data is insufficient to achieve the classification accuracy attained by the proposed approach, thereby highlighting the importance of integrating both feature types for improved tumor classification. As can be seen from Table 4.1, the proposed model which combines both MRI and WSI modalities achieves the highest accuracy.

4.5.4 Comparisons with State-of-the-Art Approaches

We now show comparisons with several state-of-the-art models in Table 4.2. Like other reported works, we have evaluated our model using three different metrics, namely, F1-Score (micro averaged), Cohen’s Kappa and Balanced Accuracy. As the

Table 4.2: Comparison with state of the art models

Models	F1 Score	Cohen's Kappa	Balanced Accuracy
Sahayam <i>et al.</i> [143]	-	-	0.754
Chan <i>et al.</i> [144]	-	-	0.78
Xue <i>et al.</i> [145]	-	-	0.849
Pei <i>et al.</i> [146]	0.886	0.801	0.8
Hamidinekoo <i>et al.</i> [147]	0.886	0.811	0.860
Wang <i>et al.</i> [163]	0.943	0.903	0.889
Ours	0.928	0.892	0.904

results demonstrate, we have clearly surpassed [143], [144], [145], [146], [147]. We have outperformed [163] in terms of balanced accuracy but are marginally losing in terms of F1 score and Cohen's Kappa.

4.6 Time Complexity Analysis

Our solution pipeline consists of three major components - VoxCNN, DenseNet-121 and GCN. The overall time complexity will be computed as the combination of the time complexities if these three components.

The architecture of VoxCNN, which is used to identify Glioblastoma from 3D MRI image, comprises of 4 CNN layers followed by 1 Fully Connected layer [155]. Let the time complexity of 3D VoxCNN model be written as-

$$T_V^{3D} = T_{conv}^{3D} + T_{fc}^{3D} \quad (4.12)$$

where T_{conv}^{3D} and T_{fc}^{3D} are given in Eq. 2.17 and 2.20 of chapter 2 respectively.

The time complexities of DenseNet-121 and GCN is defined in Sec. 2.3.3 (see Eq. 2.21) and Sec. 2.4.3 (see Eq. 2.27) respectively of Chapter 2. Combining all

the three, the overall time complexity of our pipeline can be represented as-

$$\begin{aligned}
T &= T_V^{3D} + T_{D121}^{2D} + T_{GCN} \\
&= O\left(\sum_{n=1}^4 k_{n-1} \cdot s_n^3 \cdot f_n \cdot l_n^3 \cdot r_1 \cdot b_1 + C \cdot D \cdot W \cdot H \cdot N\right) + \\
&O\left(\sum_{n=1}^{120} k_{n-1} \cdot s_n^2 \cdot f_n \cdot l_n^2 \cdot r_1 \cdot b_1 + C \cdot D \cdot W \cdot H \cdot N\right) + \\
&O(G_L \cdot G_N \cdot F^2 + G_L \cdot G_E \cdot F) \\
&= O\left(\sum_{n=1}^4 k_{n-1} \cdot s_n^3 \cdot f_n \cdot l_n^3 \cdot r_1 \cdot b_1 + \sum_{n=1}^{120} k_{n-1} \cdot s_n^2 \cdot f_n \cdot l_n^2 \cdot r_1 \cdot b_1 + 2(C \cdot D \cdot W \cdot H \cdot N)\right) + \\
&O(G_L \cdot G_N \cdot F^2 + G_L \cdot G_E \cdot F)
\end{aligned} \tag{4.13}$$

where d is the number of convolutional layers, l_n is the length of the output feature map in the n th layer, f_n is the number of filters in the n th layer, s_n is the filter size in the n th layer, k_{n-1} is the number of input channels, r_1 is the learning rate, b_1 is the batch size, C is the number of input channels, W is the width of the input, H is the height of the input, N is the number of neurons, G_L is the number of layers, G_N is the number of nodes, G_E is the number of edges and F is the number of features.

4.7 Summary

In this work, we have presented a brain tumor classification strategy by combining deep features and Graph Convolutional Networks. Use of both radiology and histopathology data is shown to achieve state-of-the-art performance. We also showed how effective graph building and embedding deep features helps Graph Convolutional Networks learn efficiently. In future, other modalities and tumor grades will be included to perform a more comprehensive classification.

Chapter 5

Detection of brain tumor related bio-markers

Brain cancer has a very high mortality rate. Gliomas are the most common malignant brain tumors, causing this severe fatality. Recent biological investigations revealed that a holistic study of bio-markers, responsible for causing genetic mutations in gliomas, can ensure a comprehensive prognosis and treatment plan for the patients. In this chapter, we simultaneously predict five such important genetic markers, namely, IDH, 1p/19q codeletion status, ATRX, MGMT, and TERT from Whole Slide Images using deep learning.

5.1 Introduction

Brain tumors account for a disproportionate burden of cancer mortality because of their high fatality rate; only one-third of individuals survive at least 5 years after diagnosis [164]. Gliomas are the most common primary malignant brain tumors in adults, arising from glial cells. More recently, several molecular signatures have been identified to predict favorable patient response to treatment as well as overall outcome in gliomas [165]. Collective study of various genetic mutations in gliomas, which are captured through different bio-markers, are quite essential for a comprehensive prognosis and subsequent treatment plan [166].

Some attempts have been made in the past to identify bio-markers from radiological data, like MRI, using machine learning techniques [167, 168]. But there are

inherent limitations to using radiological data for predicting genetic markers. Firstly, genetic markers are traditionally identified by examining histopathological data, like Whole Slide Image (WSI) samples, as the effect of genetic mutations are more visible at the cellular level [169]. Secondly, as mentioned in [170], though a model can be trained to learn the correlation between radiological features and the bio-marker status; WSI data can offer morphological features, that help a model learn better the distinguishing factors of the bio-markers. WSI offers a better explainability for models trained to identify glioma bio-markers [171]. As shown in Fig. 5.1, there are three different types of glioma tissues which exhibit quite different visual characteristics. Each of them has a different combination of bio-markers. A model can typically learn these combinations by capturing suitable features from WSI. Some related problems have been quite well investigated. For example, multi-class brain tumor classification is studied in recent times in [172, 173]. Multi-class bio-marker detection [174] and prediction [175] using WSI image have also been investigated. In such problems, one typically needs to focus on a single genetic marker, which is solely responsible for tumor mutation. However, a more challenging, and biologically more intriguing problem, is to predict multi-label genomic bio-marker, where multiple genetic markers collectively influence tumor mutation. Naturally, the complexity of the problem increases with an increase in the number of bio-markers to be detected. This class of problem remains largely unexplored.

We now discuss the biological importance of the five glioma genetic markers, which are predicted in this work. These markers are Isocitrate DeHydrogenase (**IDH**), combined loss of the short arm of chromosome 1 and the long arm of chromosome 19 (**1p/19q codeletion**), O(6)-methylguanine-DNA methyltransferase (**MGMT**), Alpha-Thalassemia/mental Retardation, X-linked (**ATRX**), and, Telomerase Reverse Transcriptase (**TERT**) in the context of gliomas. Literature review suggests that these bio-markers can affect glioma *individually*, in a *pairwise* manner, and, as a *group*. IDH genes play a pivotal role in low grade gliomas (LGG) with favorable prognosis [14]. 1p/19q codeletion leads to better prognosis in oligodendroglial tumors [176]. MGMT is a vital DNA repair protein. In glioblastoma

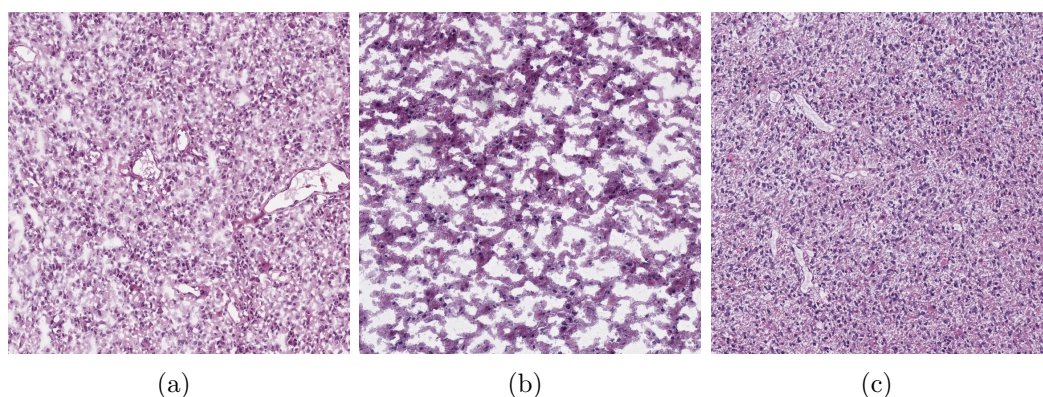


Figure 5.1: Three glioma tumor tissue samples from TCIA [9]. (a) represents a glioblastoma tumor with IDH-wildtype, No 1p/19q codeletion, ATRX-wildtype, TERT mutated and MGMT methylated . (b) represents an Oligodendroglioma tumor with IDH-mutant, No 1p/19q codeletion, ATRX-mutant, TERT unmutated and MGMT methylated. (c) represents an Astrocytoma tumor with IDH-mutant, 1p/19q codeletion, ATRX-wildtype, TERT mutated and MGMT methylated.

patients, MGMT promoter methylation status is a significant marker for therapeutic response to temozolomide, an alkylating agent [15]. Furthermore, ATRX mutations are also important as they are associated with increased survival in individuals with astrocytic tumors [177]. Finally, in the genomic landscape of gliomas, mutations in the TERT promoter region have emerged as a common occurrence [178]. As a part of pairwise behavior, one bio-marker influences the presence or absence of another bio-marker in case of gliomal tumors. For example, as mentioned in [16], 1p/19q codeletions are closely linked to IDH mutation. As demonstration of a composite behavior, multiple bio-markers are found to influence different types of glioma. For example, the presence of a TERT mutation without IDH mutation and 1p/19q codeletion suggests the occurrence of Glioblastoma [10]. Such composite behaviors are also captured by three glioma tumor tissue samples in fig. 5.1.

5.2 Related Work

We have come across some works, which have used machine learning techniques for the prediction of genomic mutations. However, most of these studies are restricted to separately predicting specific mutations only.

In this context, we first discuss the works that have used MRI data for the prediction of genetic bio-markers in gliomas. To predict the MGMT status, the authors in [179] used multi-modal MRIs, and various deep learning architectures. However, the reported results were not to the mark. In [180], the authors employed various data preprocessing techniques, and applied deep learning models to find the MGMT status. The reported results were again below par. In another work [181], radiogenomic features were extracted from MRI data to classify the 1p/19q status. Some works have attempted to predict two bio-markers from MRI data. Zhou *et al.* [182] applied a random forest approach to initially determine IDH status. They then trained a second model to predict 1p/19q status within the IDH mutant subset. Nishikawa *et al.* [183] used 3D deep learning models on MRI data to simultaneously predict the status of IDH and 1p/19q codeletion status. Tripathi and Bag [168] performed classification of tumor grade along with the determination of IDH and 1p/19q status using MRI data. But MRI data provides information and features about the tumor macro environment, which a model uses to learn the correlation between label and data [170]. Further, the tumor region needs to be segmented to learn the correlation, which requires additional processing and hence increases the computational burden. Still, the model does not really learn the features of cellular structures, which are directly changed as a result of bio-marker mutation, as shown in [169]. Thus, it fails to provide valuable information about other bio-markers like MGMT, ATRX and TERT which are analysed with WSI data in real clinical scenario. This fact is corroborated by [179], where, they did not achieve desirable prediction results for MGMT status from MRI. Similarly, as discussed in [184], the morphological effects of ATRX and TERT mutations are clearly visible in WSI images.

We also explored some works that have used WSI as their imaging modality. For IDH status prediction, Liechty *et al.* [185] used uniform manifold approximation and projection (UMAP) on patch-level embedding vectors of WSI images and achieved decent results. For the same task, Liu *et al.* [186] used multiscale feature fusion and arrived at even better results. In [14], the authors used GAN and incorporated

patient specific information like age to predict the presence of IDH mutations. Wang *et al.* [187], through their experiments using both MRI and WSI data, showed that WSI data alone provides better classification accuracy for IDH status prediction than that of from MRI. All these works clearly endorse the fact that WSI is certainly a better modality for prediction of genetic markers (as compared to MRI). The above review also indicates that there is a marked paucity of published works on simultaneous prediction of multiple mutations, which is very much essential to plan fast and efficient prognosis.

Note that the task of predicting multiple bio-markers is essentially a multi-label prediction problem. We now discuss some loss functions, which are applied for multi-label prediction tasks. Wu *et al.* [188] considered underlying distribution of data and class imbalance to improve the performance of their model. However, they used a binary cross-entropy based loss function. Further, their technique is more suitable for datasets having long-tailed distributions, which is not the case here. Ridnik *et al.* [189] employed asymmetric loss to reduce the negative effect of class imbalance. They used Regions of Interest (ROI), whereas our model learns from the whole image. Yessou *et al.* [190] compared various multi-label loss functions. However, none of these functions have incorporated any label correlation knowledge, a very important factor in medical image analysis [191].

As a summary, we can say that this is the first work which focuses on predicting five genetic markers for brain tumors from the WSI data. In our proposed deep learning based solution, we design a composite loss function, articulating different behaviors of these markers.

5.3 Contributions

In this work, we simultaneously predict five different glioma causing bio-markers, *i.e.*, IDH, 1p/19q codeletion status, ATRX, MGMT and TERT from WSI images using a deep network. A comprehensive knowledge of these five bio-markers can ensure faster and better prognosis and treatment plan for the patients. In our solution, we explicitly capture the individual, pairwise, and, group behavior of the

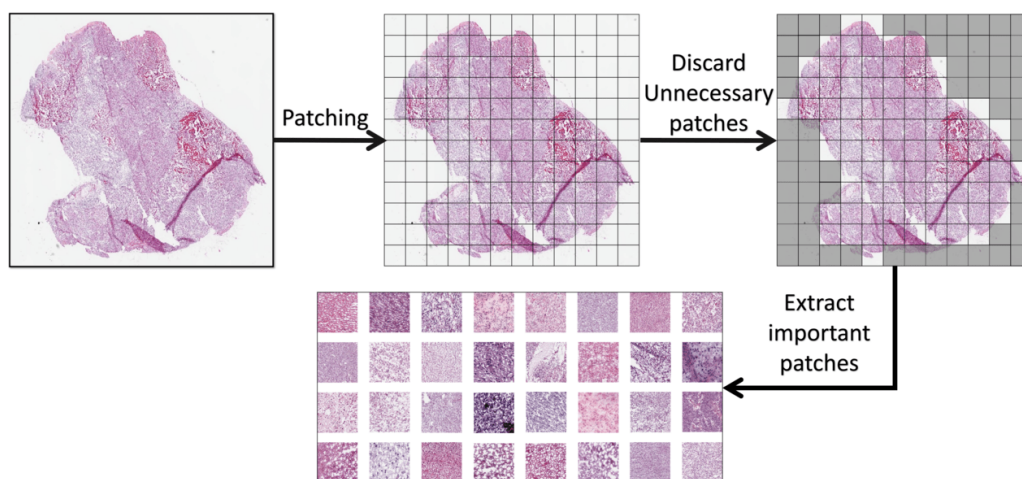


Figure 5.2: Patch generation process: A whole slide image is divided into equal sized square patches. Each patch is scanned and the uninformative patches (shown in gray) are discarded. Remaining patches containing actual tissue images are saved.

bio-markers as revealed by the existing literature. A composite loss function is designed by combining these three losses. We did not find any public dataset that has directly made available a mapping of WSI image and the corresponding bio-marker information. This mapping is crucial in order to learn label correlation information from bio-markers along with image features from WSI. We carefully analyze, clean and select data samples that had both bio-marker and WSI sample for the same patient, and, create a new benchmark dataset for ready use. Our contributions are now summarized below:

- We simultaneously predict five glioma causing bio-markers using a deep network with a composite loss function. The three components of this loss function are as follows:
 - A multi-label weighted cross-entropy loss, with weights for each bio-marker based on their probability of occurrence. We create a five-headed FCN which fine-tunes the model to calculate the multi-label weighted cross-entropy loss for each of the five individual bio-markers. This teaches the model to predict bio-markers based on their *individual traits*.
 - A conditional probability based loss function which harnesses the cause-effect relationship of the bio-marker pairs. This teaches the model to

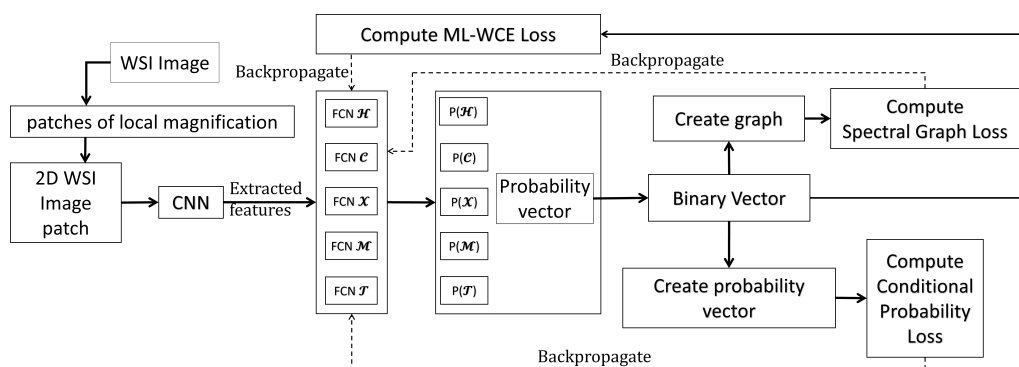


Figure 5.3: Training workflow for a single WSI image at local magnification level. This is repeated for other WSI images in the training set at local magnification level. The whole training process is then repeated at global magnification level.

predict bio-markers based on their *pair-wise relationship*.

- A graph similarity based loss function that makes the model learn the bio-marker co-occurrence information. This teaches the model to predict bio-markers based on the *composite (group-wise) relation* among them.
- We build a common benchmark data repository with WSI image and the corresponding bio-marker information by integrating WSI data from The Cancer Imaging Archive (TCIA) [9] with bio-marker data from the Genomic data commons (GDC) [192] of the same patient. We believe this type of benchmark will be beneficial for studying multiple bio-marker predictions of glioma. This dataset can be publicly accessed [here](#).
- We demonstrate through comprehensive experiments that we achieve state-of-the-art prediction results by outperforming a number of competing methods on the benchmark dataset.

5.4 Methodology

In this section, we first describe how the image is fed to the model, followed by the design of loss functions. We then discuss how we conduct the training and testing of the model. The overall training workflow of our network is shown in Fig. 5.3 whereas the testing workflow is shown in Fig. 5.4.

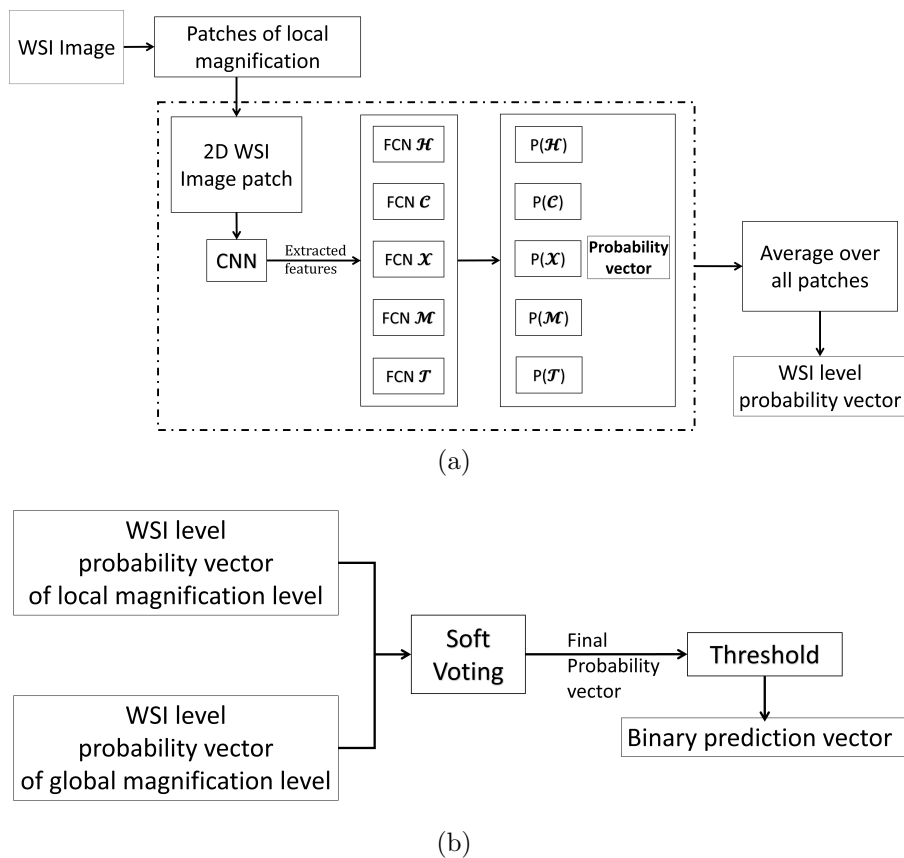


Figure 5.4: Overall test time workflow of our model. (a) shows how we get a single probability vector for a single WSI image for local magnification level. This process is repeated at the global magnification level also. (b) shows how we derive the final binary prediction from combining the results of (a).

5.4.1 Image Feature Extraction

We use a CNN to extract features from WSI. Since, a single color WSI has extremely high resolution (e.g., $\sim 20000 \times 20000 \times 3$), it becomes impractical to directly feed the image into a CNN. Hence, we divide the image into a bag of smaller equal sized patches, following [193]. Another reason to use multiple different patches from the same WSI image is to increase the sample size for training. Since, WSI is captured from the actual tissue slide, it contains several unnecessary artifacts like bubbles, dust, and, cracks. Furthermore, as the background (region without any tissue image) is white, the extracted smaller white patches are of no use. So, we discarded all the unnecessary patches using the same strategy as adopted in [194]. The patch generation process is shown in Fig. 5.2. WSI at different magnifications convey different feature information. To utilize these different types of information, patches are extracted from $20\times$ (local level) and $5\times$ (global level) magnification, similar to [195]. In the dataset, WSI is available in both magnification levels. Lower magnification levels preserve overall morphological structures, while higher magnification levels capture finer details [196]. Let each such input image patch of $20\times$ magnification of a sample WSI be denoted by I_i^l ($i = 1, \dots, n_l$), where n_l is the total number of local image patches in the bag. Similarly, each image patch of $5\times$ magnification level of the same WSI is denoted by I_i^g ($i = 1, \dots, n_g$), where n_g is the total number of global image patches in the bag. Suppose, F_i^l and F_i^g be d -dimensional feature vectors for I_i^l and, I_i^g respectively, obtained using a CNN.

The F_i^l 's and F_i^g 's are next inputted to each FCN. All the FCNs have a sigmoid layer in the end. Let the b -dimensional probability vectors denoting the occurrence of the bio-markers for I_i^l , and, I_i^g be respectively represented by p_i^l and, p_i^g . For this problem, $b = 5$. The p_i^l 's and p_i^g 's vectors are obtained from the b number of FCNs. We further construct binary prediction vectors B_i^l and B_i^g from p_i^l and p_i^g . Please see Table 5.1 for the bio-markers and the interpretation of the values of 1 and 0. For the sake of brevity, we will henceforth simply use B to denote a prediction vector. All the constituent image patches of any particular WSI sample have the same annotation (ground-truth). Hence, they have the same reference vector, say, B_r . In this way,

we have the training patches, with a prediction vector, and, a reference prediction vector for all. The goal of training the model is to produce prediction vectors B for the training patches belonging to a single WSI sample that are equal to the reference vector B_r of that WSI sample. Training is applied batch-wise on these patches. Please refer to Fig. 5.3 for the training workflow.

5.4.2 Design of Loss Function

We design three different loss functions to capture individual, pairwise and groupwise traits of the bio-markers. These three losses are eventually combined to form a composite loss.

(A) Individual Multi-Label Weighted Cross-entropy Loss

Multi-label Weighted Cross-Entropy (ML-WCE) loss assigns weights to the bio-markers based on the number of samples in the dataset for which they are present and absent. This captures impacts of the bio-markers, solely at individual level. In our dataset, the bio-markers are not evenly distributed among the samples. As this is a multi-label problem, we need to obtain separate weights for different bio-markers. Let the number of positive samples for the bio-marker i be s_i and the number of negative samples for the same be t_i . A positive sample means a value 1 for the sample, and, a negative sample means a value 0. Please see Table 5.1 for the interpretation. The probability of presence of the bio-marker i can be calculated as $w_i^s = \frac{s_i}{m}$, and, the probability of absence of the bio-marker i as $w_i^t = \frac{t_i}{m}$, where m is

Table 5.1: Details of the genetic markers: ID, Name, and, Interpretation of the values of 1 and 0 following [10].

ID	Name	Value: 1	Value: 0
\mathcal{H}	IDH	Mutant	Wild-type
\mathcal{C}	1p/19q codeletion	Codeleted	Non-codeleted
\mathcal{X}	ATRX	Mutant	Wild-type
\mathcal{M}	MGMT	Methylated	Unmethylated
\mathcal{T}	TERT	Mutant	Wild-type

the total number of WSI samples. Standard cross-entropy loss is given by:

$$\mathcal{L}_C = -[y_i \times \log(\hat{y}_i) + (1 - y_i) \times \log(1 - \hat{y}_i)] \quad (5.1)$$

where, y_i and \hat{y}_i are the actual and predicted labels for the bio-marker i respectively [197]. In this problem, we modify the above equation by adding the two weights w_i^s and w_i^t . Let us denote the modified cross-entropy loss for the bio-marker i by Ψ_i . So, we write:

$$\begin{aligned} \Psi_i = & -[w_i^s \times y_i \times \log(\hat{y}_i) + \\ & w_i^t \times (1 - y_i) \times \log(1 - \hat{y}_i)] \end{aligned} \quad (5.2)$$

Each individual Ψ_i is calculated for each individual FCN in order to learn a specific bio-marker as shown in Fig. 5.3. The loss for all the bio-markers is obtained as the average over all Ψ_i . So, we write:

$$\mathcal{L}_C = \frac{1}{b} \sum_{i=1}^b \Psi_i \quad (5.3)$$

The training process aims to minimize \mathcal{L}_C so that B_r and B becomes more similar as training progresses, with the ideal goal of making B equal to B_r .

(B) Pairwise Conditional Probability Loss

We next formulate a conditional probability based loss function to model pairwise behavior of the bio-markers. The conditional probability links the presence/absence of one bio-marker caused by presence/absence of a second bio-marker.

Let the estimated conditional probability be denoted by $\mathcal{P}(i = \tau_1 | j = \psi_1)$, where, i, j are the bio-markers ($i = 1, \dots, b; j = 1, \dots, b; i \neq j$). Let the actual conditional probability, which is given as ground-truth, be denoted by $\mathcal{P}_r(i = \tau_2 | j = \psi_2)$. Here, $\tau_1, \tau_2, \psi_1, \psi_2 \in \{0, 1\}$. We show in the Appendix with an example how \mathcal{P} can be obtained from B . Likewise, one can obtain \mathcal{P}_r from B_r . The conditional probability loss is deemed as the absolute difference between the sum of estimated conditional

probabilities and actual conditional probabilities and, can be expressed as:

$$\mathcal{L}_P = \sum_{i=1}^b \sum_{j=1}^b \left(\left| \mathcal{P}_r(i = \tau_1 | j = \psi_1) - \mathcal{P}(i = \tau_2 | j = \psi_2) \right| \right. \\ \left. \times \delta_{\tau_1, \tau_2} \times \delta_{\psi_1, \psi_2} \right) \quad (5.4)$$

In the above equation, δ_{τ_1, τ_2} is equal to 1 if τ_1 and τ_2 are equal, and is 0 otherwise. Similar is the interpretation for δ_{ψ_1, ψ_2} . For the present problem, we selected 5 sets of conditional probabilities, which are rendered biologically important by the experts. As mentioned in [198], TERT mutations commonly occurs with 1p/19q codeletion, therefore the probability $\mathcal{P}(\mathcal{T} = 1 | \mathcal{C} = 1)$ is high and is an important relation to consider. Similarly, as mentioned in [16], 1p/19q co-deletions are mutually exclusive with ATRX mutations, *i.e.*, $\mathcal{P}(\mathcal{C} = 1 | \mathcal{X} = 0)$ and $\mathcal{P}(\mathcal{C} = 0 | \mathcal{X} = 1)$ are important relations. We likewise formulate the rest of the pairwise relations based on evidences given in [10, 16]. Hence, for this work, the general equation in (5.4) can be simplified to:

$$\mathcal{L}_P = \left| \mathcal{P}_r(\mathcal{T} = 1 | \mathcal{H} = 1) - \mathcal{P}(\mathcal{T} = 1 | \mathcal{H} = 1) \right. \\ + \mathcal{P}_r(\mathcal{M} = 1 | \mathcal{H} = 1) - \mathcal{P}(\mathcal{M} = 1 | \mathcal{H} = 1) \\ + \mathcal{P}_r(\mathcal{C} = 1 | \mathcal{X} = 0) - \mathcal{P}(\mathcal{C} = 1 | \mathcal{X} = 0) \\ + \mathcal{P}_r(\mathcal{C} = 0 | \mathcal{X} = 1) - \mathcal{P}(\mathcal{C} = 0 | \mathcal{X} = 1) \\ + \mathcal{P}_r(\mathcal{T} = 1 | \mathcal{C} = 1) - \mathcal{P}(\mathcal{T} = 1 | \mathcal{C} = 1) \\ \left. + \mathcal{P}_r(\mathcal{X} = 1 | \mathcal{H} = 0) - \mathcal{P}(\mathcal{X} = 1 | \mathcal{H} = 0) \right| \quad (5.5)$$

where, $\mathcal{H}, \mathcal{C}, \mathcal{X}, \mathcal{M}$ and \mathcal{T} refer to the bio-markers as shown in Table 5.1. The training process aims to minimize \mathcal{L}_P .

Now, we show how to calculate the probability $\mathcal{P}(i = \tau_1 | j = \psi_1)$ for any two bio-markers τ and ψ . To obtain $\mathcal{P}(i = \tau_1 | j = \psi_1)$, we need the predicted binary vector B . Since, we get a batch of B s as output during training, we can calculate $\mathcal{P}(i = \tau_1)$, $\mathcal{P}(j = \psi_1)$ and $\mathcal{P}(i = \tau_1 \cap j = \psi_1)$ by first finding the number of times $i = \tau_1$ occurs, $j = \psi_1$ occurs, and, $i = \tau_1$ and $j = \psi_1$ jointly occur, respectively; and

then dividing each by the batch size. An example of how to find $\mathcal{P}(i = 1|j = 0)$ is presented below.

Let us consider a batch with size $z_1 + z_2 + z_3 + z_4$. Here, z_1 is the number of occurrences where both i and j have values 1; z_2 is the number of occurrences where i has a value 1 but j has a value 0; z_3 denotes the number of cases where i has a value 0 but j has a value 1; and, z_4 is the number of cases where both i and j have values 1.

$$\mathcal{P}(j = 0) = \frac{z_1 + z_2}{z_1 + z_2 + z_3 + z_4} \quad (5.6)$$

$$\mathcal{P}(i = 1 \cap j = 0) = \frac{z_2}{z_1 + z_2 + z_3 + z_4} \quad (5.7)$$

Hence,

$$\mathcal{P}(i = 1|j = 0) = \frac{\mathcal{P}(i = 1 \cap j = 0)}{\mathcal{P}(j = 0)} = \frac{z_2}{z_1 + z_2} \quad (5.8)$$

(C) Group-wise Spectral Graph Loss

We introduce here a third type of loss using a graph-based formulation. This loss aims to make the model learn about co-occurrence of multiple bio-markers. The group information of the five genetic markers is captured through a weighted complete graph $G = G(V, E)$. Each vertex of this graph denotes a genetic marker as explained in Table 5.1. So, $V = [v_1, v_2, \dots, v_b]$ and $|V| = b$. We previously obtain the b -dimensional binary prediction vector, B . This means, $B[i]$ denotes the value of v_i and likewise. In the complete graph, we represent the affinity of the bio-markers by assigning appropriate weights to the edges. For example, in a particular sample, if three out of the five bio-markers occur together, then the edges connecting them are assigned high values. In contrast, the edges connecting these vertices with the two remaining vertices are given low values. This is shown in Fig. 5.5, where, three different graphs are built from the genetic information of the sample images shown

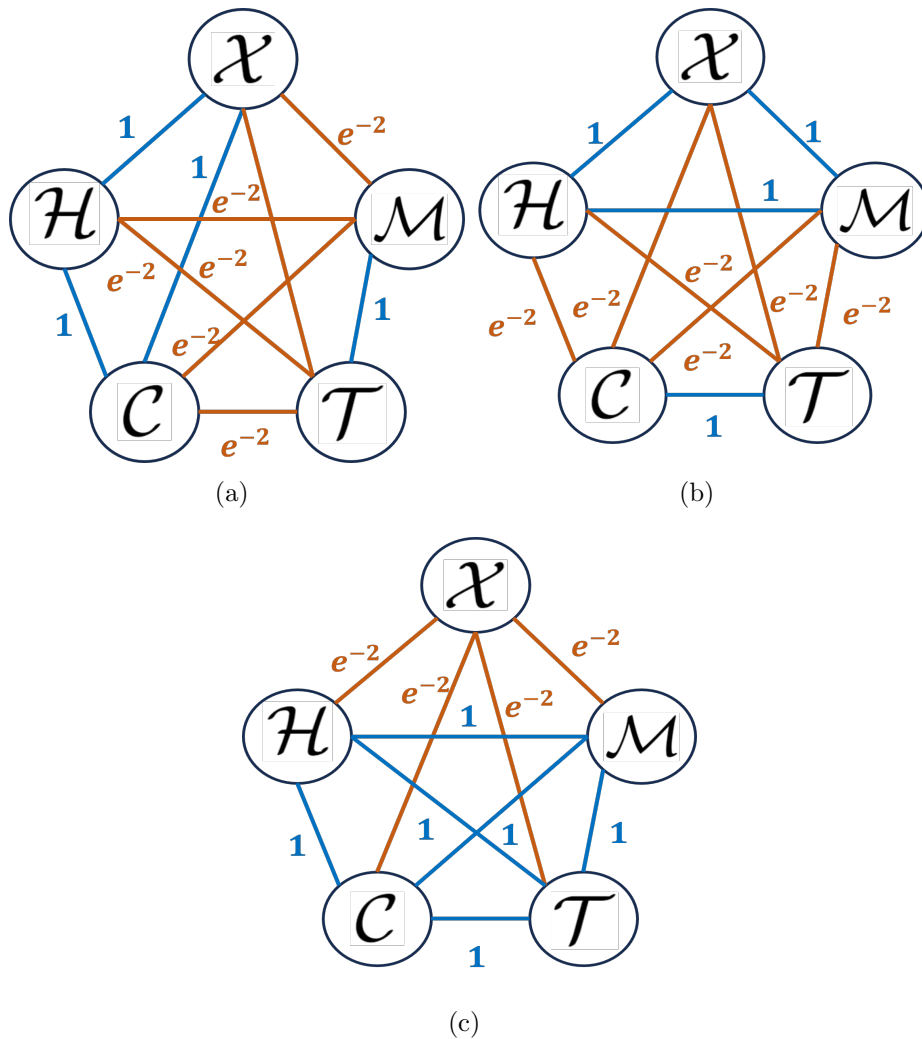


Figure 5.5: Graphs in left, middle, and right are for the three glioma tumor tissue samples shown in Fig. 5.1. Here, the vertices are the status of bio-markers, an edge with weight 1 connects two bio-marker vertices having same value as denoted in Table 5.1. An edge with weight e^{-2} means the bio-markers connected by that edge have different values.

in Fig. 5.1. We set the edge weights (w_{ij}) between the vertices v_i and v_j using the following equation:

$$w_{i,j} = e^{-2|v_i-v_j|} \quad (5.9)$$

So, $|v_i - v_j|$ can assume a value of either 0 (when v_i and v_j have same values) or 1 (when v_i and v_j have opposite values). Exponentiation is introduced to provide more discrimination between these two cases, making the edge weights either 1 (when v_i and v_j have same values) or e^{-2} (when v_i and v_j have opposite values). This WSI sample has a reference ground-truth binary vector B_r (provided by the expert annotators in the dataset). So, one can create a complete reference graph G_r from B_r in the same manner as G is built from B . The procedure to create graph G from binary vector B is described below.

Let a predicted vector be represented by $\hat{B}_I^J = [0, 1, 1, \dots, 0, 1]$ (where I and J have their usual meanings). This sample also has a ground truth represented by $B_I^J = [0, 1, 0, \dots, 1, 1]$. Let \hat{G}_I^J and G_I^J be the graphs constructed from \hat{B}_I^J and B_I^J respectively. The procedure to create graph G_I^J from binary vector B_I^J is described below.

To ensure that nodes having same values are connected together and to make the whole process compatible with back propagation, *i.e.*, to make each step derivable, we use matrix based operations like negation and stacking of 1-D vectors to create a 2-D adjacency matrix A .

1. First we find $B' = 1 - B$ which is the complement of a binary vector B .
2. Then, we create A by concatenating either B or B' row-wise.
3. If, in the original vector B , the i^{th} index is 1, then we add B as a row in A .
Otherwise, we add B' as a row in A .

Algorithm 5.1 describes the above process in a concise manner. The goal of the training process is to minimize the structural difference between the two graphs G and G_r . This difference is calculated using spectral decomposition of graphs [199]. Let A and A_r be the adjacency matrices of G and, G_r respectively. Further, let D

and D_r be the diagonal degree matrices of G and, G_r respectively. Then, we can write:

$$\begin{aligned} L &= I - D^{-1/2}AD^{1/2} \\ L_r &= I - D_r^{-1/2}A_rD_r^{1/2} \end{aligned} \quad (5.10)$$

where, L and L_r are respective symmetric normalized Laplacian of the graphs, and I is the identity matrix. We then compute the eigenvalues (λ) of the above Laplacians, and, define the spectral graph loss as:

$$\mathcal{L}_G = \sum_{i=1}^k |\lambda_{(i)} - \lambda_{r(i)}|^2 \quad (5.11)$$

where k is given by the following equation:

$$\begin{aligned} k = \min \left[\left\{ \arg \min_q \left(\frac{\sum_{i=1}^q \lambda_i}{\sum_{i=1}^b \lambda_i} \right) > \phi \right\}, \right. \\ \left. \left\{ \arg \min_t \left(\frac{\sum_{i=1}^t \lambda_{r(i)}}{\sum_{i=1}^b \lambda_{r(i)}} \right) > \phi \right\} \right] \end{aligned} \quad (5.12)$$

Here, ϕ is a threshold. We keep the top k eigenvalues that contain $(\phi \times 100)\%$ of the energy. Two graphs, G and G_r are more similar if \mathcal{L}_G is closer to 0. So, the goal is to minimize \mathcal{L}_G .

(D) Composite Loss

We have formulated three different loss functions to make the model learn three different types of information. The total loss is designed as a linear combination of the three loss functions and can be hence expressed as:

$$\mathcal{L}_T = \alpha \mathcal{L}_C + \beta \mathcal{L}_P + \gamma \mathcal{L}_G \quad (5.13)$$

The three constants, α , β and γ appearing in the above equation, are determined experimentally.

Algorithm 5.1: Training Process

Input: n_l number of local image patches each denoted as I_i^l
Output: b -dimensional binary prediction vector B

```

1 while loss is not minimized do
2    $i \leftarrow 0$ 
3   while  $i < n_l$  do
4     Extract local features  $F_i^l$  from  $I_i^l$  using a CNN
5     Compute  $b$ -dimensional probability vectors and  $p_i^l$  by inputting  $F_i^l$ 
      into  $b$  number of FCNs respectively
6     Compute  $b$ -dimensional binary prediction vectors  $B_i^l$  from and  $p_i^l$ 
      respectively
7     Compute ML-WCE Loss from Eq. 5.3 using  $B_i^l$ 
8     Compute Conditional Probability Loss from Eq. 5.5 using  $B_i^l$ 
9     Compute Spectral Graph Loss from Eq. 5.11 using  $B_i^l$ 
10    Backpropagate losses
11     $i \leftarrow i + 1$ 
12  end
13 end
14 return  $B$ 

```

5.4.3 Training Phase

After the WSI images are processed as described in 5.4.1, we have two sets of 2D image patches, one belonging to $20\times$ magnification and another belonging to $5\times$ magnification. We train two models having exactly the same architecture as shown in Fig. 5.3 for each of these two sets. In each model, deep features are first extracted by a CNN, which are fed into b FCNs. Each FCN is trained to predict the status of a specific bio-marker using the ML-WCE loss. Output probabilities from each FCN are first concatenated and then binarized to produce a binary vector. This binary vector is further used in the conditional probability based loss and spectral graph loss. The losses are then back-propagated to the CNN and FCNs. The spectral graph loss is calculated per sample as done in standard neural network architectures [158, 200] whereas the conditional probability loss is calculated per batch (see Appendix). The entire training workflow for $20\times$ magnification, as shown in Fig. 5.3, is then repeated for $5\times$ magnification. The algorithm for $20\times$ magnification is given below, which is the same for $5\times$ magnification. The steps in the training phase are summarized in Algorithm 5.1.

5.4.4 Testing Phase

During the testing phase, we extract patches at the local and global magnification levels from a WSI image in exactly similar manner as during the training phase. We need a single prediction for each of the b bio-markers from a WSI sample (of a patient data). So, we first average the prediction probabilities of each local patch to obtain the prediction vector for the test WSI [193]. Likewise, we average the probabilities of each global patch to obtain the probability vector for the same test WSI. We finally apply a weighted soft voting approach to combine the two results to get a single binary output probability vector [201]. The weights are set automatically using a modulated rank averaging technique calculated from the training accuracies of local and global level models [202]. The testing workflow is shown in Fig. 5.4, where, we predict a value of 1 or 0 for each bio-marker from a test WSI image. Algorithm 5.2 lists the steps for the testing phase.

Algorithm 5.2: Testing Process

Input: n_l number of local image patches each denoted as I_i^l , n_g number of global image patches each denoted as I_i^g

Output: b -dimensional binary prediction vector B

```

1  $i \leftarrow 0$ 
2  $j \leftarrow 0$ 
3 while  $i < n_l$  and  $j < n_g$  do
4   Extract local and global features  $F_i^l$  and  $F_i^g$  from  $I_i^l$  and  $I_i^g$  respectively
   using a CNN
5   Compute  $b$ -dimensional probability vectors  $p_i^l$  and  $p_i^g$  by inputting  $F_i^l$ 
   and  $F_i^g$  into  $b$  number of trained FCNs respectively
6   Compute  $b$ -dimensional binary prediction vectors  $B_i^l$  and  $B_i^g$  from and  $p_i^l$ 
   and  $p_i^g$  respectively
7    $i \leftarrow i + 1$ 
8    $j \leftarrow j + 1$ 
9 end
10 Average all  $B_i^l$  and  $B_i^g$  to get  $B^l$  and  $B^g$  respectively
11 Perform soft voting on  $B^l$  and  $B^g$  to get  $B$ 
12 return  $B$ 

```

5.5 Experiments

5.5.1 Data Preparation

In order to predict the presence or absence of genetic bio-markers from WSI, an image to label correspondence is required. The Cancer Imaging Archive (TCIA) [9] has Hematoxylin and Eosin (H&E) stained WSI images of patients from the TCGA-GBM [203] and TCGA-LGG [204] datasets. The bio-marker information is available from the cBioPortal repository [192]. We mapped the WSI images from TCIA with the genetic bio-marker information for the same patient. The details of the data preparation are presented below.

(A) Genomic Data selection and processing

Clinical and genomic information about the patients of TCGA-GBM and TCGA-LGG dataset are available from the TCGA PanCancer Atlas dataset in the cBioPortal [192]. We selected samples with information available on specifically five bio-markers, the names and different values for which are shown in Table 5.1.

(B) WSI Data selection and processing

WSI data is available in .svs format. Each file contains multiple zoom levels, capturing the H&E stained tissue from various magnifications. Due to the very high-resolution of WSI images, for example, $\sim 20000 \times 20000 \times 3$ in our dataset, rendering and subsequent training of deep learning algorithms on this data becomes unfeasible even with latest Graphics Processing Units (GPUs). Therefore, we adopted a patch-based approach inspired by the work of [193]. We divided each WSI into smaller patches of resolution $224 \times 224 \times 3$. We automatically selected relevant patches using a histogram based patch selection technique as explained in Sec. 5.4.1. The overall patch generation and selection process is shown in Fig. 5.2.

(C) Data Aggregation

We did some checks before aggregating the data. Firstly, we removed multiple instances of the same patient from both the datasets. We selected only the latest

Table 5.2: Table showing the counts of each type of genetic marker. The values and ID are similar to those in the Table. 5.1

ID	Value: 1	Value: 0
\mathcal{H}	360	330
\mathcal{C}	133	557
\mathcal{X}	164	526
\mathcal{M}	472	218
\mathcal{T}	371	319

versions of samples from both datasets. Secondly, in the cBioPortal, we found that not all patients had information about all the five bio-markers. We discarded samples having more than 2 bio-marker data missing. For samples that had one or two bio-marker missing, we took help from neurologists to derive the status of those missing bio-markers, considering the type of tumor and the status of other bio-markers. Finally, we aggregated WSI data from TCIA and bio-marker data from cBioPortal having same patient/subject ID which satisfy the above criteria. A repository with one-to-one correspondence is created, where, each subject ID has one WSI image and a list (binary vector) of five elements containing information about the presence/absence of bio-markers associated with it. The total number of final samples is 690, among which 388 are female and 302 are male patients. The cancer type distribution in the dataset is as follows, Glioblastoma Multiforme - 275, Oligodendroglioma - 226 and Astrocytoma - 189. The individual counts of the bio-markers are given in Table 5.2.

(D) Dataset division

We randomly divided the cohort of 690 patients into three sets. Specifically, 483 patients (70%) were allocated for training, 69 (10%) for validation, and the remaining 138 (20%) were set aside for testing purposes.

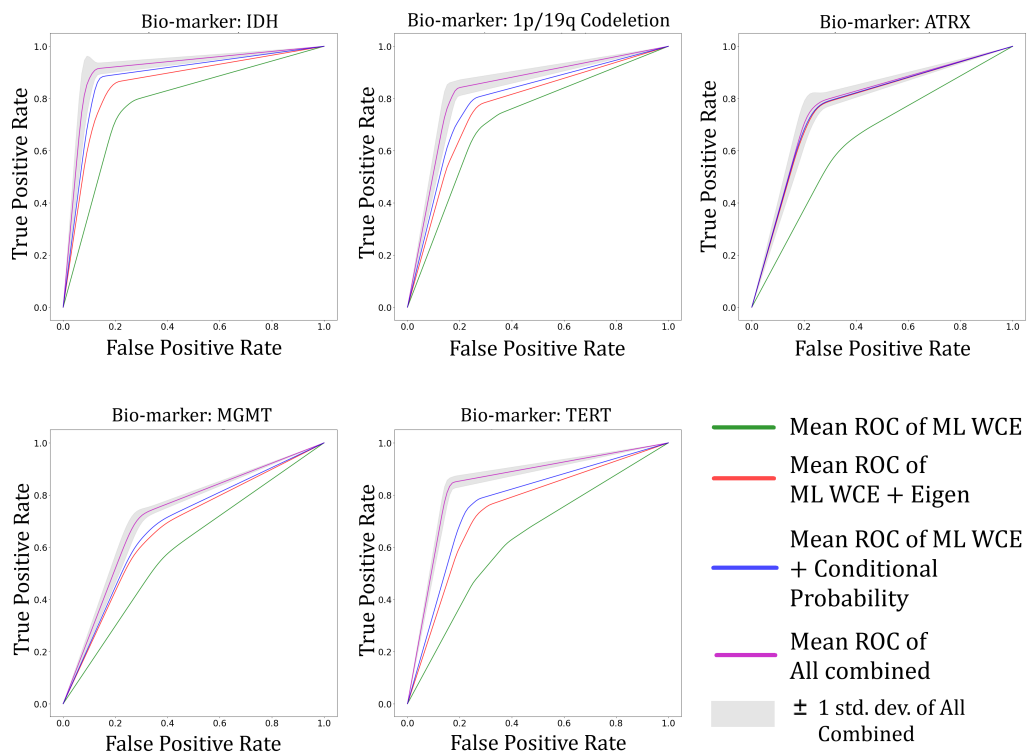


Figure 5.6: The AUC for each bio-marker achieved by the models in shown in Table 5.4. The gray bands represent 95% confidence interval for the AUC of our proposed model. The mean ROC of ML-WCE, ML-WCE + Spectral Loss, ML-WCE + Conditional Probability Loss and the mean ROC of all combined is shown with green, red, blue and magenta lines respectively.

5.5.2 Experimental Settings

(A) Network Architecture

We use Kimianet [205] as the CNN for extracting histopathological features. It is pre-trained on a vast number of histopathology images and has better prediction performance than other popular CNNs like DenseNet [158] and EfficientNet [200]. The FCN is made with fully connected layers and dropout, followed by a sigmoid activation function in the end. We use dropout layers to prevent over-fitting, and each dropout layer randomly selects 10% of the nodes in the preceding layer and makes their output 0. The final out has shape - batch size \times 1 as the final output is a binary prediction indicating the status of a bio-marker. It is passed through a sigmoid activation function. The FCN architecture is shown in Table 5.3.

(B) Loss function parameters

We set $\phi = 0.9$ in (Eq. 5.12), *i.e.*, we keep the top k eigenvalues that have 90% of the energy. The other parameters (Eq. 5.13) are used to control the weights (impact) of each component of the final loss function. We tried different combinations of α , β and γ values where each of them can have values belonging to the set $[0.1, 0.2, 0.3, \dots, 1]$ similar to the procedure followed in [206]. After experimentation, we have found that $\alpha = 0.5, \beta = 0.2, \gamma = 0.3$ give optimum results.

Table 5.3: Layer architecture of FCN. Each dropout layer drops 10% of the neurons randomly.

Layer	Input shape	Output shape
Linear	batch size x 1280	batch size x 640
Dropout (0.1)	batch size x 640	batch size x 640
Linear	batch size x 640	batch size x 256
Dropout (0.1)	batch size x 256	batch size x 256
Linear	batch size x 256	batch size x 64
Dropout (0.1)	batch size x 64	batch size x 64
Linear	batch size x 64	batch size x 1

(C) Hyperparameters

The learning rate and the batch size were respectively set to 3×10^{-4} and 256. We used the Adam optimizer for optimizing the weights and trained our model for 200 epochs. We further set $b = 5$, $m = 690$, $n_g = 200$ and $n_l = 2000$.

(D) Computing platform and performance metrics

All the computations are done using an HP Z640 Workstation with Intel Xeon 14-core Processor, 128GB RAM and NVIDIA Titan RTX 24GB Graphics processor. We employed PyTorch 1.9 [161] on an Ubuntu 20.04 platform. Considering the above hardware, the time taken to predict bio-markers from a single WSI sample is 10.0 minutes on average. This includes the time taken to split a WSI into its corresponding patches, running inference, and then combining the patch inference results to get the final prediction (for the whole sample). The average inference time for a single image patch is around 0.2 seconds.

Similar to most reported works on mutation prediction using histopathology images ([182, 207, 208]), we have employed AUC, Sensitivity, and Specificity as the performance measures in this work. We additionally computed 95% confidence intervals (CI) for the AUC.

5.6 Results

During the training phase, individual image patches are utilized to train the model. For consistency, the same approach is followed during the testing phase. WSI of a subject is partitioned into smaller patches. The patch level probability vector outputs are aggregated and binarized into a WSI level binary vector, which serves as the final prediction.

5.6.1 Ablation Study

We have conducted an ablation study to reveal the impacts of individual loss functions on the prediction of each bio-marker. We have calculated the Area Under Curve (AUC) values along with 95% CI for each bio-marker. The AUC is derived

Table 5.4: Ablation Study: Impacts of individual (ML-WCE) loss, pairwise (Conditional Probability based) loss, and groupwise (Spectral Graph based) loss

Loss Function	IDH			1p/19q Codeletion			ATRX			MGMT			TERT			Mean AUC
	AUC with 95% CI	Sens.	Spec.	AUC with 95% CI	Sens.	Spec.	AUC with 95% CI	Sens.	Spec.	AUC with 95% CI	Sens.	Spec.	AUC with 95% CI	Sens.	Spec.	
ML-WCE	0.78 (0.729 - 0.812)	0.781	0.77	0.717 (0.636 - 0.771)	0.717	0.715	0.643 (0.556 - 0.715)	0.64	0.618	0.593 (0.5 - 0.654)	0.594	0.592	0.63 (0.518 - 0.763)	0.627	0.631	0.672
ML-WCE + Spectral Graph Loss	0.85 (0.800 - 0.912)	0.85	0.834	0.767 (0.706 - 0.859)	0.767	0.768	0.77 (0.726 - 0.824)	0.77	0.75	0.669 (0.587 - 0.751)	0.67	0.663	0.745 (0.677 - 0.822)	0.745	0.743	0.761
ML-WCE+ Conditional Probability Loss	0.88 (0.859 - 0.912)	0.87	0.88	0.793 (0.729 - 0.859)	0.792	0.79	0.772 (0.726 - 0.824)	0.773	0.764	0.682 (0.609 - 0.751)	0.68	0.65	0.773 (0.718 - 0.821)	0.778	0.77	0.78
Composite Loss = ML-WCE + Spectral + Conditional Probability Loss	0.913 (0.879 - 0.932)	0.883	0.831	0.845 (0.812 - 0.878)	0.793	0.757	0.783 (0.735 - 0.815)	0.732	0.72	0.72 (0.696 - 0.752)	0.684	0.658	0.849 (0.829 - 0.867)	0.783	0.762	0.823

Table 5.5: Comparison of various loss functions for prediction of five genetic markers

Method	IDH			1p/19q Codeletion			ATRX			MGMT			TERT			Mean AUC
	AUC	Sens.	Spec.	AUC	Sens.	Spec.	AUC	Sens.	Spec.	AUC	Sens.	Spec.	AUC	Sens.	Spec.	
Hamming Loss [209]	0.764	0.713	0.723	0.717	0.683	0.672	0.643	0.617	0.623	0.598	0.572	0.581	0.738	0.724	0.731	0.692
SparseMax Loss (SML) [210] (2016)	0.847	0.84	0.791	0.798	0.748	0.723	0.73	0.71	0.684	0.631	0.632	0.62	0.729	0.687	0.664	0.747
Asymmetric Loss [189] (2019)	0.83	0.735	0.724	0.793	0.774	0.752	0.73	0.698	0.68	0.69	0.676	0.651	0.778	0.752	0.738	0.7642
Distribution Based Loss [188] (2020)	0.8	0.79	0.795	0.735	0.734	0.736	0.692	0.693	0.689	0.609	0.596	0.581	0.717	0.702	0.697	0.7106
Ours	0.91	0.878	0.81	0.835	0.77	0.73	0.778	0.72	0.71	0.719	0.68	0.63	0.845	0.75	0.75	0.817

from the Receiver Operating Characteristic (ROC) curve, which is plotted by varying the decision threshold of our model output and then plotting the True Positive Rate (TPR) vs. False Positive Rate (FPR) at each threshold. In the first model, we use only the ML-WCE loss function. This serves as a base model. In the second model, we combine ML-WCE loss, and, Spectral Graph Loss. The third model replaces Spectral Graph Loss with conditional probability loss. In the final model, all three loss functions are combined. As can be seen from Table 5.4, the proposed model, which combines all the loss functions, achieves the highest AUC for all five bio-markers compared to the previous three combinations. We also show the mean AUC for each model. It is important to note that each loss function individually improves the mean AUC from the base ML-WCE model, which indicates that each loss function contributes significantly to the overall improvement. Fig. 5.6 further conveys this message in a qualitative manner. From the figure, we can further conclude that MGMT prediction is challenging, as all the models performed below par in predicting MGMT.

5.6.2 Comparison of Loss Functions

We have compared our loss function with other reported works on multi-label image prediction using different loss functions. Although, these works used nonmedical

Table 5.6: Comparison with state-of-the-art. AUC, Sensitivity (Sens.) and Specificity (Spec.) values are given, as per availability. Best values are highlighted in bold.

Method	IDH			1p/19q Codeletion			ATRX			MGMT			TERT			Mean	
<i>Single bio-marker prediction</i>																	
	AUC	Sens.	Spec.	AUC	Sens.	Spec.	AUC	Sens.	Spec.	AUC	Sens.	Spec.	AUC	Sens.	Spec.	AUC	
Liu <i>et al.</i> [186] (2024)	0.886	0.793	0.755														
Liu <i>et al.</i> [14] (2020)	0.927	0.889	0.813														
Liechty <i>et al.</i> [185] (2022)	0.870	0.76	0.709														
Kocak <i>et al.</i> [181] (2019)				0.838	0.877	0.675											
Faghani <i>et al.</i> [180] (2023)										0.65	0.712	0.589					
Saeed <i>et al.</i> [179] (2022)										0.53							
Mora <i>et al.</i> [211] (2023)							0.831	0.772	0.697								
Ours	0.913	0.883	0.831	0.845	0.793	0.757	0.783	0.732	0.72	0.72	0.684	0.658	0.849	0.783	0.762		
<i>Two bio-markers prediction</i>																	
Zhou <i>et al.</i> [182] (2019)	0.916	-	-	0.716	-	-	-	-	-	-	-	-	-	-	-	-	0.816
Nishikawa <i>et al.</i> [183] (2023)	0.814	-	-	0.771	-	-											0.7925
Tripathi and Bag [168] (2023)	0.919	0.909	0.934	0.878	0.895	0.915											0.8985
Ours	0.913	0.883	0.831	0.845	0.793	0.757											0.879
<i>Five bio-markers prediction</i>																	
Xiao <i>et al.</i> [212] (2023)	0.812	0.784	0.767	0.736	0.712	0.698	0.685	0.675	0.691	0.673	0.614	0.605	0.793	0.755	0.741	0.74	
Weng <i>et al.</i> [213] (2023)	0.913	0.854	0.817	0.823	0.807	0.813	0.772	0.746	0.712	0.697	0.641	0.623	0.824	0.795	0.786	0.8058	
Densenet [158]	0.71	0.7	0.72	0.683	0.62	0.61	0.56	0.57	0.524	0.499	0.48	0.52	0.657	0.58	0.53	0.6218	
Efficientnet (b0) [200]	0.739	0.74	0.71	0.631	0.68	0.623	0.6	0.64	0.59	0.557	0.58	0.547	0.588	0.64	0.6	0.623	
Ours	0.913	0.883	0.831	0.845	0.793	0.757	0.783	0.732	0.72	0.72	0.684	0.658	0.849	0.783	0.762	0.823	

data, the modality is still 2D images like ours. Apart from a generic multi-label loss function, like Hamming loss [209], we re-implemented Asymmetric Loss [189], Distribution based Loss [188] and the best performing loss function - SpareMax Loss (SML) [210] from [190]. Table 5.5 shows that our proposed approach outperforms the above works, thereby further reinforcing the need of developing a loss function, which can faithfully capture all the intricacies like individual, pairwise, groupwise behavior of the bio-markers.

5.6.3 State-of-the-art Comparisons

In Table 5.6, we have compared our work with state-of-the-art works on bio-marker prediction. Note that most of these methods predicted only one bio-marker. We first compare our multi-label prediction results, which is much more challenging against works that have done single bio-marker prediction. Among these works, most of them have attempted to predict IDH mutation status because this is one of the most crucial bio-markers for glioma tumor diagnosis. It is true that Liu *et al.* [14] marginally outperformed us in IDH detection using WSI data. Nevertheless, it is important to mention that a binary prediction task is easier than that of a multi-label prediction. Kocak *et al.* [181] attempted to identify the 1p/19q codeletion status in

which we outperform them in terms of AUC and specificity. We found two works that attempted MGMT status prediction. Out of these two methods, we outperform Saeed et al.'s work [179] by a huge margin. We lag slightly behind sensitivity of MGMT status prediction of [180], but outperform them on AUC, which represents the overall performance of any model. Although Mora *et al.* [211] surpasses our ATRX detection, they incurred additional computational cost in model training for brain tumor segmentation followed by radiological features from it. Since, we have used WSI, no such extra computation is required. We did not find any work that attempted to identify TERT mutation status, which is an important bio-marker for tumor prognosis. It is natural to expect that the multi-label performance would not be able to exceed single bio-marker prediction performance. But for proper diagnosis and prognosis of brain tumors, information on multiple bio-markers is collectively required [214], which renders the attempt to predict the status of just one bio-marker unsuitable for a real clinical scenario.

We then list some works that have attempted to identify the status of two bio-markers - IDH and 1p/19q codeletion. We did not find any works that have tried to delve into simultaneous prediction of the status of other bio-markers like ATRX, MGMT and TERT. Zhou *et al.* [182] targeted prediction of two bio-markers - 1p/19q and IDH. Here, we lose in the IDH prediction extremely marginally; but, our overall mean is quite better in comparison. Nishikawa *et al.* [183] also attempted to predict the same set of bio-markers, we outperformed them by a big margin. Tripathi and Bag [168] obtained better results in both IDH and 1p/19q codeletion status prediction. However, their method is not suitable for prediction of other bio-markers, as discussed in Sec. 5.2. Also, they had to segment the tumor region first before attempting to identify bio-markers as only the tumor region has features related to the correct bio-marker status of the patient while the rest of the healthy brain region has a different bio-marker status and learning from the entire brain scan would thus, confuse the model. Segmenting the tumor required additional computational burden and processing times, which is not needed in WSI based prediction models like ours.

Finally, we compared our work on five-class multi label prediction with some

state-of-the-art methods on multi-label prediction. Since, to the best of our knowledge, our work is the only work that targets simultaneous multi-label prediction of IDH, 1p/19q codeletion, TERT, ATRX and MGMT status, we had to compare our method with similar methods but solving a different problem. We re-implemented Xiao et al.'s work [212] that performs a similar task of multi-label classification on our dataset and found that the performance is not up to the mark as they have focused on learning image features without paying attention to the label correlation. We also compared our work with another work that focus on label correlation [213] by re-implementing it on our dataset and found that it lags behind our results. We have further included comparisons with other baseline models, namely, DenseNet and EfficientNet, both of which are pre-trained with ImageNet [215] and fine-tuned only in the last FCN layer using a binary cross-entropy loss function. The results clearly establish the supremacy of our method. Overall, we can see that our proposed method outperforms multi-label prediction models and also a few single label prediction models. We have also achieved the highest mean AUC value.

5.7 Time Complexity Analysis

In this section, we analyze the time complexity of our proposed method. Our solution pipeline consists of mainly three components - the deep network consisting of a CNN backbone connected to five FCNs from which the weighted cross entropy losses are calculated, the conditional probability loss calculation unit and the spectral graph loss calculation unit. The complexity analysis focuses on the dominant computational operations involved in their calculations.

5.7.1 Time complexity of CNN and FCNs

We have used Kimianet [205] as the CNN backbone which is based on the DenseNet-121 architecture. Using Eq. 2.16 defined in Sec. 2.3.1 of Chapter 2, we can derive the time complexity for a 2D DenseNet-121 model having 120 convolutional layers

as-

$$T_{DCNN} = O \left(\sum_{n=1}^{120} k_{n-1} \cdot s_n^2 \cdot f_n \cdot l_n^2 \cdot r_1 \cdot z \right) \quad (5.14)$$

where d is the number of convolutional layers, l_n is the length of the output feature map in the n th layer, f_n is the number of filters in the n th layer, s_n is the filter size in the n th layer, k_{n-1} is the number of input channels, r_1 is the learning rate, z is the batch size.

We use five FCNs after that. The time complexity of 2D FCN has already been defined in Sec. 2.3.2 (See 2.19). Hence, the cumulative time complexity of 5 FCNs each having 3 FC layers can be written as-

$$T_{DFCN} = O \left(5 \cdot \sum_{l=1}^3 C \cdot W \cdot H \cdot N \right) \quad (5.15)$$

where C is the number of input channels, W is the width of the input, H is the height of the input, N is the number of neurons. Thus, combining T_{DCNN} and T_{DFCN} , the overall time complexity for the deep learning model can be written as-

$$\begin{aligned} T_{net} &= T_{DCNN} + T_{DFCN} \\ &= O \left(\sum_{n=1}^{120} k_{n-1} \cdot s_n^2 \cdot f_n \cdot l_n^2 \cdot r_1 \cdot z \right) + O \left(5 \cdot \sum_{l=1}^3 C \cdot W \cdot H \cdot N \right) \\ &= O \left(\sum_{n=1}^{120} k_{n-1} \cdot s_n^2 \cdot f_n \cdot l_n^2 \cdot r_1 \cdot z + 5 \cdot \sum_{l=1}^3 C \cdot W \cdot H \cdot N \right) \end{aligned} \quad (5.16)$$

5.7.2 Complexity of Conditional Probability Loss ($\mathcal{L}_{\mathcal{P}}$)

The key computational steps in calculating the conditional probability loss are as follows-

1. Computation of Conditional Probabilities:

Given b biomarkers, the number of possible biomarker pairs is $O(b^2)$. For each pair, conditional probabilities are estimated from the dataset, which involves counting occurrences over m samples. This step has a computational complexity of $O(b^2m)$.

2. Loss Computation:

The loss function is defined as the absolute difference between estimated and ground-truth probabilities. Since only a fixed number of biologically significant biomarker pairs (denoted as c) are selected, this step has a complexity of $O(c)$, where $c \ll b^2$.

If all possible pairs of biomarkers were considered, the worst-case complexity would be $O(b^2m)$. However, given that only a constant number of pairs (c) are selected based on expert recommendations, the overall complexity simplifies to:

$$O(m) \tag{5.17}$$

Thus, the computational cost of the conditional probability loss is primarily dependent on the number of samples in the dataset.

5.7.3 Complexity of Spectral Graph Loss (\mathcal{L}_G)

The major steps in computing the spectral graph loss are as follows:

1. Graph Construction

A complete weighted graph G is constructed from the predicted biomarker values, where each vertex represents a biomarker, and edges encode biomarker affinities. The adjacency matrix A of size $b \times b$ is computed using Eq. (5.9), leading to a complexity of $O(b^2)$.

2. Graph Laplacian Computation:

The degree matrix D and normalized Laplacian matrix $L = I - D^{-1/2}AD^{1/2}$ are computed. Constructing the degree matrix requires summing the rows of A ($O(b^2)$), while obtaining the normalized Laplacian matrix involves matrix multiplications, leading to a complexity of $O(b^3)$.

3. Eigenvalue Decomposition: The top k eigenvalues of the Laplacian matrix are computed, where k is chosen such that it captures a fixed proportion (ϕ) of the spectral energy. Eigenvalue decomposition for an $b \times b$ matrix has a complexity of $O(b^3)$.

4. Loss Computation:

The spectral loss is computed as the squared difference between the top k eigenvalues of the predicted and ground-truth graphs. This step has a complexity of $O(k)$, where $k < b$.

The dominant term in the complexity analysis is the eigenvalue decomposition step, which has a computational complexity of:

$$O(b^3) \tag{5.18}$$

Thus, the spectral graph loss is computationally more expensive than the conditional probability loss, particularly as the number of biomarkers (b) increases.

5.7.4 Overall Time Complexity

Combining the time complexities of CNN, FCN, conditional probability loss and spectral graph loss, we arrive at the final time complexity of our pipeline below-

$$\begin{aligned} T_{final} &= T_{net} + O(m) + O(b^3) \\ &= O\left(\sum_{n=1}^{120} k_{n-1} \cdot s_n^3 \cdot f_n \cdot l_n^3 \cdot r_1 \cdot z + 5 \cdot \sum_{l=1}^3 C \cdot W \cdot H \cdot N\right) + O(m) + O(b^3) \\ &= O\left(\sum_{n=1}^{120} k_{n-1} \cdot s_n^3 \cdot f_n \cdot l_n^3 \cdot r_1 \cdot z + 5 \cdot \sum_{l=1}^3 C \cdot W \cdot H \cdot N + m + b^3\right) \end{aligned} \tag{5.19}$$

5.8 Summary

This chapter introduces a novel approach to simultaneously predict five genetic markers—IDH, 1p/19q codeletion status, ATRX, MGMT, and TERT—from WSI data, improving prognosis and treatment planning for brain tumors. Our method leverages individual, pairwise, and groupwise traits of these bio-markers, integrating conditional probability-based loss for pairwise information, spectral graph-based loss for group behavior, and multi-label weighted cross-entropy loss for individual traits

into a composite function. We also created a benchmark dataset combining WSI and bio-marker data, which shows state-of-the-art performance in experiments.

Chapter 6

Alzheimer’s Disease Classification

The development of computer vision techniques for early Alzheimer’s diagnosis is essential for detecting subtle brain changes at preclinical stages, enabling timely interventions that can delay cognitive decline and improve patient outcomes. These techniques offer a scalable, non-invasive, and cost-effective solution, making widespread screening feasible and advancing research in neurodegenerative diseases. Ultimately, this work is critical in the global effort to mitigate the impact of Alzheimer’s Disease.

6.1 Introduction

Alzheimer’s disease (AD) is the most common type of dementia, accounting for about 60% to 70% of the total number of dementia cases in the World. This deadly disease is caused by the damage and destruction of nerve cells in the brain regions related to memory, and its most common symptoms are memory loss and cognitive decline. Worldwide, around 50 million people have dementia, and there are nearly 10 million new cases every year. Globally, the current number of AD and dementia patients is at least 55 million. At this rate, it could exceed 152 million by 2050 [216]. According to data from [217], the number of older persons — those aged 65 years or over — is expected to more than double by 2050 and to more than triple by 2100, rising from about 700 million globally in 2019 to 1.8 billion in 2050 and 2.45 billion in 2100 as shown in Fig. 6.1. Currently, there is no treatment available to cure AD or to alter its progressive course. However, in order to support and improve the quality

of lives of AD patients, the treatment process has a big scope of improvement. Early detection and diagnosis is a major goal for dementia and AD care. AD has generally three major stages of progression-

1. Early Mild Cognitive Impairment (EMCI)
2. Late Mild Cognitive Impairment (LMCI)
3. Alzheimer's Disease (AD)

Along with the above three AD stages, we have one more class - Normal Control (CN), *i.e.*, those who have no symptoms of AD, LMCI or EMCI. So the main task is to classify a brain scan into one of the four major AD classes. Previous efforts of classification were mainly focused on binary classification, *i.e.*, to develop algorithms to classify between any two classes, for example, between AD and CN, between LMCI and CN, between EMCI and CN, and so on. Although it was easy to differentiate among only two types of data, but it was more time-consuming to determine the actual class as one would have to eliminate each class by making multiple binary comparisons until the final class is reached. In this work, we for the first time address a direct four-class classification problem related to AD. clearly, this task is significantly more challenging than single or multiple binary classification(s) as the differences among all four classes is not very distinct.

For the diagnosis of dementia, use of Magnetic Resonance Imaging (MRI) has been quite common. MRI can create a 3D representation of the internal brain structure through magnetic fields and radio waves, It offers the possibility of in-vivo study of pathological brain changes associated with AD. Currently, Diffusion weighted MRI (DWI or DW-MRI) is the standard version of MRI which is used in clinical diagnosis worldwide. It uses the diffusion of water molecules to generate contrast in MR images. Building upon DWI, DTI has gained popularity as it is able to capture the directions of water molecule diffusion, thus providing a lot more data about brain tissue structure than conventional MRI [218]. This extra information from DTI scans have opened avenues for a wide range of neurological applications, detecting AD is one of them.

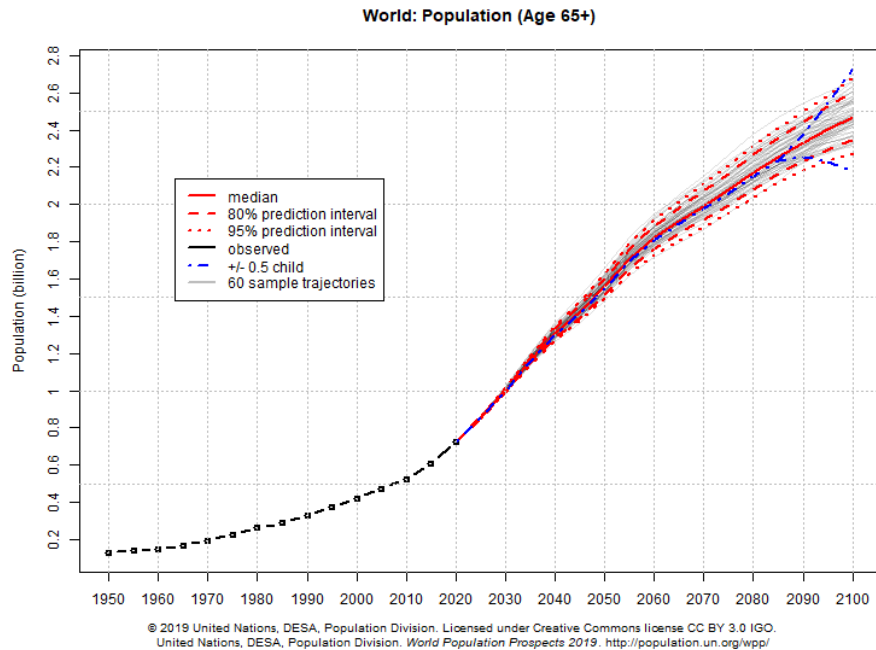


Figure 6.1: Estimates and probabilistic projections of population of people aged 65 years and above in the world

Over the past decades, neuroimaging data have been used to characterize AD by exploiting machine learning (ML) methods, offering valuable tools for diagnosis and prognosis [219, 220]. Many studies have proposed the use of predefined features (including regional and voxel-based measurements) obtained from image preprocessing pipelines along with different types of classifiers, namely, support vector machines (SVM) or random forests. In recent times, deep learning (DL), a more advanced and complex ML methodology, has created a compelling effect in the domain of medical imaging. The main advantage of DL over ML is that it allows the automatic abstraction of low-to-high level latent feature representations (e.g. lines, dots or edges for low level features, and objects or complex shapes for high level features). Compared with 2D convolutions on slices, 3D convolutions on a whole MRI can capture potential 3D structural information, which may be essential for discrimination and has proved to be advantageous on AD vs. Mild Cognitive Impairment (MCI) classification [221]. Also, it is important to note that datasets collected in neuroimaging studies are generally very small, compared to the large number of images available in datasets for image classification which are currently used to train

neural networks for object classification and detection in 2D image analysis. This leads to relatively lower accuracies, as there are not enough training examples for the network to learn the required features. This is mitigated by various techniques like data augmentation, feature fusion and decision fusion.

In this chapter, we present a solution for the above four-class AD classification problem by combining DL and ML models on 3D DTI scan volumes. The DTI data is publicly available at the Alzheimer's Disease Neuroimaging Initiative (ADNI) website, which has been discussed elaborately in section 6.6.1. As we demonstrate in the chapter, different types of data existing in DTI can be efficiently utilized when separate DL and ML models are applied. We use 3D-CNN, specifically VoxCNN, as the DL model in this work as it has shown good performance on MRI data [155]. We employ three VoxCNNs to train three types of 3D volumetric data, namely, Echo Planar Imaging, Fractional Anisotropy and Mean Diffusivity in each DTI scan. For the ML part, we apply a random forest classifier to classify derived metadata in the form of region-averaged Fractional Anisotropy and Mean Diffusivity values. Outputs from all the four models, i.e., three VoxCNNs and the random forest classifier, are finally combined with a modulated rank averaging decision fusion approach.

6.2 Related Work

Use of medical imaging for solving classification problems has been popular for a long time. Classification is critical for clinicians to ensure proper diagnosis of a disease. Neuroscience has many such applications of classification, where neuroimaging has been extensively used. Some prominent examples include brain tumor classification [222], texture classification in ALS disease [223] and AD classification [22, 224]. As stated earlier, our solution for the four-class AD classification is based on fusion of DL and ML techniques. In this section, we mainly discuss the prominent DL and ML methods and the associated fusion strategies for solving such problems. Shortcomings of the existing strategies are also analyzed, revealing the importance of our proposed algorithm.

6.2.1 Machine Learning (ML) methods

We first discuss about the existing ML techniques that have attempted to classify AD. Extracted bio-markers and processed medical descriptors, together with statistical and conventional ML methods, have been widely used to aid the classification of AD. In [225], the authors have extracted 83 Regions of Interest (ROI)s from 3D brain MRI and Positron-emission tomography (PET) scans, and proposed a Multi-fold Bayesian Kernelization (MBK) to diagnose AD. A Support Vector Regression (SVR) based decision support system was developed by Bucholz *et al.* [226] which achieved a good accuracy. We use classical ML approach like Random Forest Classifier for only a part of our proposed method, as it has limited feature extraction capabilities. However, a part of the DTI data is best handled by the Random Forest Classifier as the data contains numerical features with overlapping and very close values (linearly inseparable) making it less effective for Support Vector Machines. Further, since this part of the data has no spatial correlation, DL methods are not suitable for handling them.

6.2.2 Deep Learning (DL) Methods

Now, we discuss certain DL methods which have been widely used in computer-aided diagnosis literature. Ramaniharana *et al.* [227] used 2D slices of MRI scans to perform AD classification based in eigen values, parametric and non-parametric classifiers. Raza *et al.* [228] used a mix of ML and DL approaches to classify AD. Although, ROI-based and 2D slice-based methods can efficiently extract relevant features and partly reduce the feature dimension, they are too empirical to capture the critical features which are associated with AD classification. In contrast, 3D-CNN can capture more complete spatial features through its space association capability. Cheng *et al.* [229] extracted a number of 3D patches from the whole MRI and transformed those patches into features by 3D-CNN. Finally, multiple 3D-CNNs were used to combine the features yielding better results for AD classification which inspired us to use DTI data similarly. We also took inspiration from the work of [230] where they used Random forest ensembles to detect AD and predict progression from MCI

to AD. We used Random forest for training, a type of derived data obtained from the DTI scans. Although supervised DL methods work better than the unsupervised methods, they are highly data dependent. The performance of the model depends on the number of training data available and in most neuroimaging domains, the number of data is found to be insufficient to make a model highly accurate and effective. To address this challenge we found DTI data to be useful as each scan has 3 3D volumetric images increasing the number of data three-fold and also derived information about each brain region is available which can also be utilized using machine learning methods for classification task.

6.2.3 Data fusion techniques

Data fusion has been used from a long time since 1990s [231]. Gaining improved information with less error possibility in detection or prediction and superior reliability can be considered as the goal of fusion. Similar or multi-modal data can be fused in many ways and such strategies have been shown to be very effective in particular cases of intrusion and malware detection, prediction in traffic systems, pattern recognition, face identification, and in biomedical domains including analysis of genetic subgroups in inherent AD [232]. Data fusion technology has improved with the advent of machine learning algorithms, as shown in the survey [233]. There are mainly two types of data fusion, namely, feature fusion and decision fusion. Feature fusion is the fusion of features already available in the data or fusion of deep features extracted from different DL models. Decision fusion, on the other hand, marks fusion of decisions, *i.e.*, outputs of different models, in such a way that the final output is more accurate than the individual model outputs. Fusion is mainly done in cases where there are multi-modal data available for the same object at the same time. Many DL networks have been developed for such purpose like [234] where the models are first trained for each type of data and later fused to obtain an output. We follow a similar strategy. However, we have combined DL and ML models on separate information in DTI data and designed a new decision level fusion strategy in this work.

As a summary, we can say that AD classification works till date deal with binary classification problems and are mostly restricted to using MRI or PET data. Due to less number of available data and less information in MRI or PET data, satisfactory accuracy for multi-class classification could not be achieved. By exploiting multiple types of data available in DTI, creating appropriate DL and ML models for them and finally fusing the individual outputs at decision level, we have obtained in this work state-of-the-art performance for a four-class AD classification problem.

6.3 Contributions

Our main contributions are now summarized below:

1. We address for the first time a direct four-class classification problem in AD, which is significantly more challenging than the pre-existing single or multiple binary classification(s).
2. We efficiently harness the full potential of DTI scans by applying appropriate DL and ML models in 3D for different types of information existing in them. This detailed exploration of DTI data and that too in 3D is largely absent in the prevalent literature.
3. Finally, outputs of each learning model are combined at the decision level using a modulated rank averaging technique, thereby achieving state-of-the-art classification accuracy.

6.4 Basics of Diffusion Tensor Imaging

DTI is able to capture white-matter (WM) tracts in the brain while MRI is only limited to grey-matter (GM) visualizations. Alterations in WM diffusivity on DTI are known to be associated with clinical disease severity starting from the pre-clinical stages of AD. The WM integrity on DTI and its importance has been discussed in details in [235]. Hence DTI is a preferred choice for studying AD as it can capture both GM and WM information.

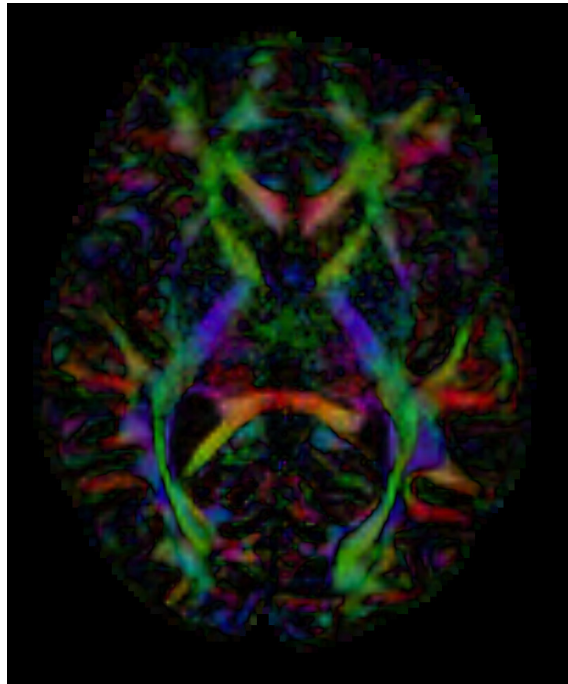


Figure 6.2: Axial FA map of a DTI scan from the ADNI dataset.

We now explain briefly about its mathematical motivation. DTI is a sensitive probe of cellular structure that works by measuring the diffusion of water molecules [27] inside living tissues. Since the diffusion tensor is a symmetric 3×3 matrix, it can be described by its eigenvalues $(\lambda_1, \lambda_2, \lambda_3)$ and eigenvectors (e_1, e_2, e_3) . The eigenvalues and eigenvectors are then used to process scalar indices and, in some studies, tractography analysis [236]. At each voxel, the eigenvalues represent the magnitude of diffusion and the corresponding eigenvectors reflect the directions of maximal and minimal diffusion.

Generally, each DTI scan contains Echo Planar Imaging (EPI) volume in 3D, and some diffusion tensor information at each voxel which can be used to generate Fractional Anisotropy (FA) and Mean Diffusivity (MD) volumes which are also in 3D. In the dataset that we used, FA and MD volumes were already available along with the EPI volume. The voxel intensities acquired from the MRI scan are stored in the EPI volume. The two main diffusion indices, FA and MD, are based on the eigenvalues, which represent the magnitude of the diffusion process.

6.4.1 Mean diffusivity

MD is a summary measure of the average diffusion properties of a voxel and is equivalent to the estimated Apparent Diffusion Coefficient (ADC) over three orthogonal directions [237]. In other words, it is a measure of the mean water diffusion rate. MD values of the voxels differ for brain scans belonging to different classes of Alzheimer's and also differ in normal healthy brains. An increase in MD indicates decreased myelination and loss of axons [238]. It can be mathematically represented as-

$$MD = \frac{\lambda_1 + \lambda_2 + \lambda_3}{3} = \frac{D_{xx} + D_{yy} + D_{zz}}{3} = \frac{Trace}{3} \quad (6.1)$$

where D_{xx}, D_{yy}, D_{zz} are the diagonal terms of the diffusion tensor.

6.4.2 Fractional anisotropy

FA is a normalized measure of the fraction of the tensor's magnitude due to anisotropic diffusion, corresponding to the degree of anisotropic diffusion or directionality and ranges from 0 (isotropic diffusion) to 1 (anisotropic diffusion). Just like MD, decreased FA values are indicative of dementia and Alzheimer's. FA values are rotationally invariant, i.e. they do not have any orientation information. It can be mathematically expressed as-

$$FA = \sqrt{\frac{3}{2}} \sqrt{\frac{(\lambda_1 - D)^2 + (\lambda_2 - D)^2 + (\lambda_3 - D)^2}{\lambda_1^2 + \lambda_2^2 + \lambda_3^2}} \quad (6.2)$$

where $D = \frac{\lambda_1 + \lambda_2 + \lambda_3}{3}$. FA maps are color coded where a certain direction is represented by a color. In these maps, red color represents left-to-right orientation, green posterior-to-anterior and blue inferior-to-superior diffusion as shown in Fig. 6.2 which has been generated with the 3D Slicer tool (version 4.10.2) [239, 240].

6.4.3 Echo Planar Imaging

EPI is the fastest imaging sequence currently available and has the potential to revolutionize many aspects of MRI technology. It is a rapid MRI technique that is capable of producing tomographic images at video rates. It is a single-shot method

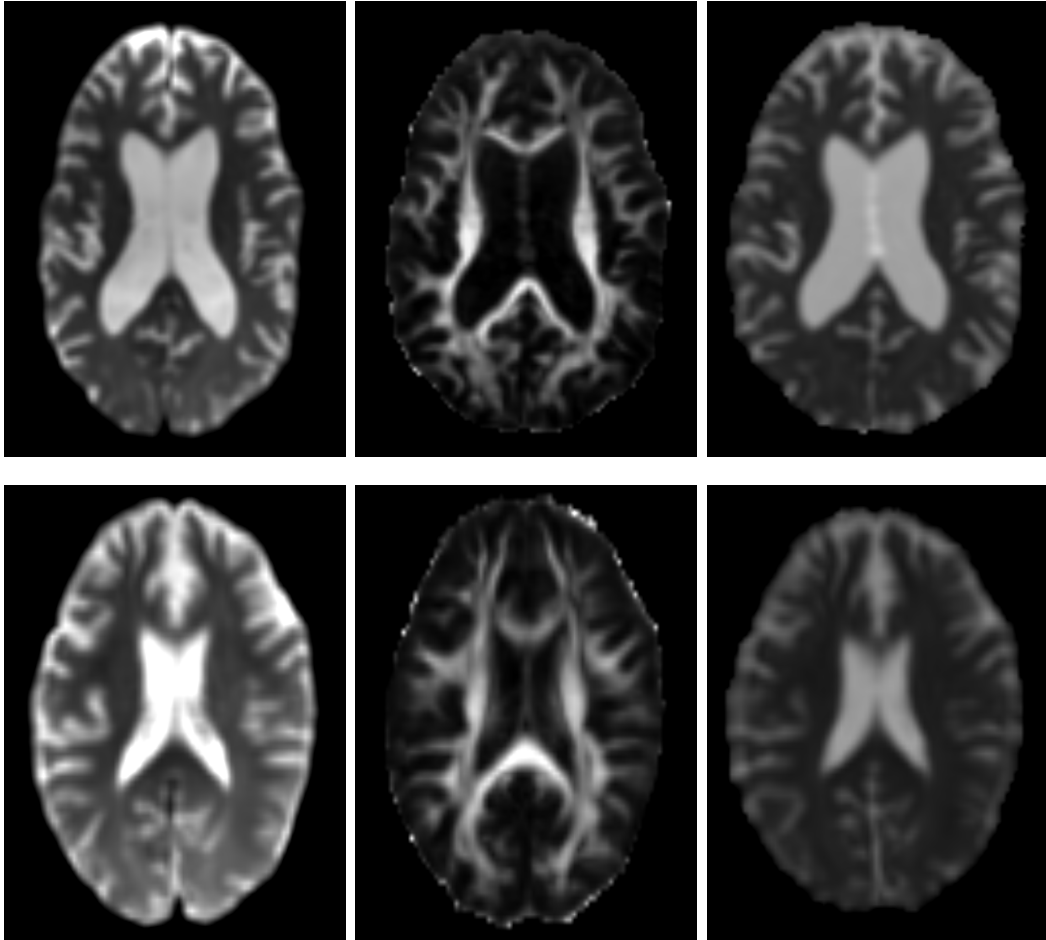


Figure 6.3: Axial slices of two patients. (a), (b) (c) denote the EPI, FA and MD maps slices of AD patient respectively and (e), (f), (g) denote that of healthy person. EPI and MD shows the WM in black, while FA shows the WM in white. We can see that in AD patient the total WM region is lesser than normal person which is clearly indicated by the larger size of lateral ventricle body in AD patient

having imaging times $\sim 100ms$ for a 128×128 matrix.

In a single-shot echo planar sequence, the entire range of phase encoding steps, usually up to 128, are acquired in one shot. In multi-shot echo planar imaging, the range of phase steps is equally divided into several shots. Each subsequent echo results in a progressively T2-weighted signal. We use DTI images with this EPI standard in our experiments.

6.4.4 Significance of Fractional Anisotropy and Mean Diffusivity

AD results in the loss of neurons in the brain and this neuronal degeneration can be seen as a loss of both GM and WM. Loss of neurons in certain areas of the brain results in GM atrophy that can be measured on conventional MR images. Increased MD was consistently found in the areas such as the hippocampus, the entorhinal cortex, the parahippocampal gyrus, the temporo-parietal association cortex, and the posterior cingulate gyrus [241].

Majority of the published AD research has used a cross-sectional design and consistently revealed low FA and high MD in widespread WM regions including the frontal, parietal, and temporal lobes (including hippocampal regions), as well as the corpus callosum and longitudinal association tracts) [238]. Thus FA and MD are clear biomarkers and their values indicate which stage of AD the patient is in. This can be seen in Fig. 6.3 where EPI, FA and MD slices of two different patients, one with AD and one healthy, are shown with visible difference in brain WM structures. These unique traits of DTI have motivated us to explore this imaging modality for a four-class classification of AD. So, we decide to apply and combine appropriate classifiers on the DTI data to achieve the above goal.

6.5 Proposed Method

Our solution pipeline consists of three VoxCNN networks and one random forest classifier, each of which outputs a 4×1 probability vector. Each probability vector contains the probabilities for the data to be in the four classes, namely, AD, CN,

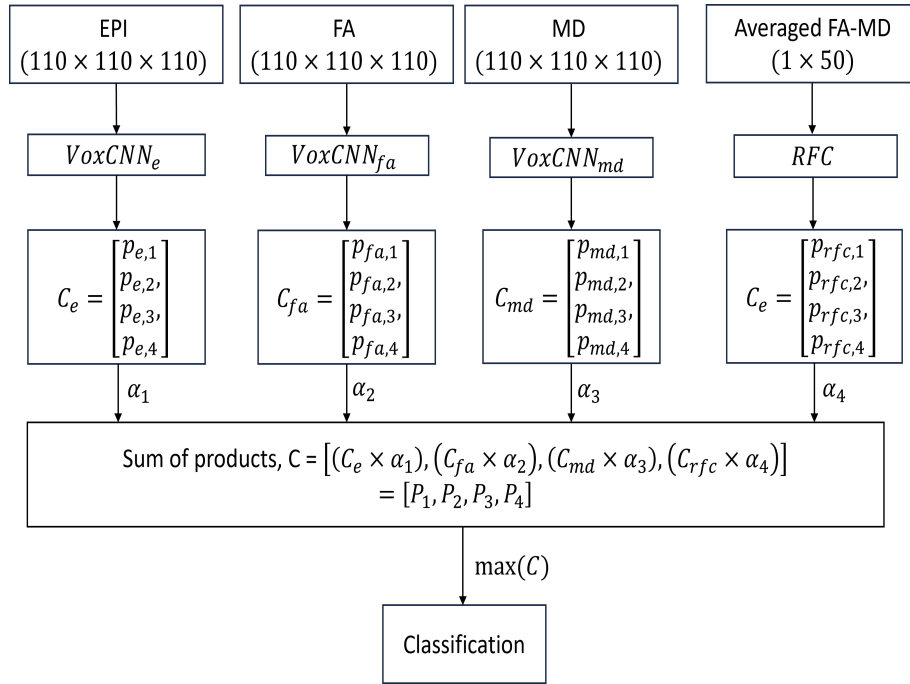


Figure 6.4: Solution pipeline architecture: C_e, C_{fa}, C_{md} and C_{rfc} denote the probability vectors of EPI, FA, MD and RFC models respectively, each of which contains the probabilities of the four AD classes; Finally, P_i s denote the probabilities for the four AD classes after applying modulated rank averaging. $\max(C)$ gives the highest value among the P_i s which corresponds to the AD classes. For example, if the output of $\max(C)$ is P_2 , then the AD class corresponding to label 2 is the classification output.

EMCI and LMCI. The probability vectors are then linearly combined by ranking the models based on their accuracy. In particular, we multiply each vector with a weight which is proportional to the rank of the corresponding model and the difference in accuracy between the corresponding model and the model which is rank-wise immediately next to it. We present a schematic of our solution in Fig. 6.4. An algorithm showing the details of the fusion strategy is shown in Algorithm 6.1. We end this section with another algorithm (Algorithm 6.2) where all the steps of our solution are mentioned. We now provide detailed descriptions of our DL models, ML models and the modulated rank averaging technique for decision level fusion.

6.5.1 VoxCNN

VoxCNN architecture has four volumetric convolutional blocks for extracting features (with a number of filters increasing from layer to layer), two deconvolutional layers with batch normalization and dropout for regularization and an output with SoftMax nonlinearity for classification. We have kept the network architecture as it is defined in the article [155]. The input files are in nifti format, and they are normalized between 0 and 1 for keeping similarity among all models. This data pre-processing part has been done with the nilearn and Scikit-learn packages [242, 243]. Considering the dataset size, model size and the limitations of GPU memory, we modified the batch iteration process in order to get samples of each class in every batch. The reason being that the probability of having only one class represented inside a batch for infinite number of samples is $\frac{1}{c^b}$ where c is the number of classes and b is the batch size. Therefore, for large batch sizes this probability is low. However, for our problem, it is high enough to thwart the learning process. Balancing of the samples inside each batch was hence undertaken to obtain stable learning curves.

6.5.2 Random Forest Classifier

Random forest classifier (RFC), an ensemble of decision trees [74] is chosen for its very high accuracy and capability to handle large volume of data. Random forests

are well suited for multi-class classification, they do not tend to overfit, can handle outliers well and has fewer number of parameters to tune [244] as compared to other state of the art classifiers like Gradient Boosting Machines (GBM) [245], XGBoost [246], etc. Also, they are resilient to outliers and clinical data is often prone to outliers [247]. Each DTI scan comes with averaged FA values of 57 regions and averaged MD values of the same 57 regions, which is derived from the images and is not actually image data. Hence, we can say that we are performing classification based on meta data in tabular format. Although neural networks can perform classification on tabular data, it is more complex in terms of setting up the model and hyper-parameter tuning, and also it is computationally expensive, whereas RFCs do not need a lot of memory resources and the training can be parallelized in a multi-core processor that greatly speeds up the training process which is a very crucial factor in the field of medical imaging where online and in-situ measurements are indispensable [248]. In order to use all available information from each DTI scan and considering the suitability of DL for volumetric image data and RFC for tabular meta data, we have used RFC along with VoxCNN to arrive at the best possible outcome. Each feature value in the meta data is a real number in $[0, 1]$. Thus, RFC is employed (code available at [243]) to randomly select a subset of features from the total feature set to arrive at a suitable classification decision. RFC votes for the most popular class among the individual trees. The information gain I for the j^{th} node in a decision tree is given by-

$$I = H(S_j) - \sum_{i=L,R} \frac{|S_j^i|}{|S_j|} H(S_j^i) \quad (6.3)$$

where $H(S_j)$ denotes the entropy of the j^{th} node S_j . The entropy of a node for a discrete set of C labels $c = 1, 2, \dots, C$ is given by:

$$H(S_j) = - \sum_{c \in C} p(c) \log_2(p(c)) \quad (6.4)$$

Further, $|S_j|$ denotes the number of training images in the node S_j . So, $|S_j^L|$ and $|S_j^R|$ respectively represent the number of data in the left child and the right child

of the node S_j . However, before the data is fed into the RFC, it goes through a set of pre-processing steps described below.

(A) Synthetic Minority Oversampling Technique

Since the data is class imbalanced, Synthetic Minority Oversampling Technique (SMOTE) sampling [249] has been used which generates excellent synthetic samples from the data and re-samples all classes to match the number of samples in the majority class.

Let x_{ij} be the value of j^{th} variable ($j = 1, \dots, p$) for the i^{th} sample ($i = 1, \dots, n$) that belongs to class c ($C = 4$) in our case. Let, $k_c = \frac{n_c}{n}$ is the proportion of samples from Class c , n_c is the number of samples in class c and n is the total number of samples. Further, let the sample size of the minority class be denoted by n_{min} . We would limit our attention to $G \leq$ variables that are the most informative about the class distinction. We use Capital letters (as X) to denote random variables, lowercase letters (as x) to denote observations and bold letters (\mathbf{x}) to indicate a set of variables. The Gaussian distribution with mean μ and standard deviation σ is indicated with $N(\mu, \sigma)$ and the uniform distribution is defined on $[0, 1]$ with $U(0, 1)$.

For each sample from the minority class (x), 40 (or $n_{min} - 1$ if $n_{min} \leq 5$) samples from the minority class with the smallest Euclidean distance from the original sample were identified (nearest neighbors) and one of them was randomly chosen (x^R). The new synthetic SMOTE sample was defined as-

$$\mathbf{S} = X + u \cdot (\mathbf{x}^R - \mathbf{x}) \quad (6.5)$$

where u was randomly chosen from $U(0, 1)$. Note that u is same for all variables, but different for each SMOTE sample.

(B) Spatially Uniform ReliefF

Spatially Uniform ReliefF (SURF) identifies the nearest neighbors (both hits and misses) based on a distance threshold from the target instance defined by the average distance between all pairs of instances in the training data. Since, there are 114

continuous valued features in the data which are very similar to each other, having values between 0 and 1, it is quite challenging to classify the data. Hence, SURF [250] is used to select the top 50 best features among the 114 features, which improves the accuracy of classification.

6.5.3 Modulated Rank Averaging based decision level fusion

In traditional majority voting method, the prediction results of majority of the classifiers are used as the final prediction labels. Since, each classifier is independent and the error rates between different classifiers are irrelevant, such strategy can be useful. However, for multi-class classification tasks, this method may not be very effective. Single classifiers perform well on most subjects; but for some subjects which are difficult to classify, the error rates will be increased due to the uncertainty among multiple categories. Let us take the following example with three classifiers. The four-class output probabilities from the SoftMax layer of these three classifiers for {AD, EMCI, LMCI, CN} are respectively given by I: {0.7, 0.05, 0.2, 0.05}, II: {0.3, 0.5, 0.1, 0.1}, III: {0.2, 0.4, 0.3, 0.1}. Based on the majority voting method, the prediction result is EMCI. But this inference is not completely correct, since the prediction result of classifier I is more credible (the winner is predominant) while that of classifiers II and III have more uncertainties (differences between the winner class and other classes are not that high).

In our approach, we use a weight adjusted probability vector fusion technique along with ranking of the classification models based on their individual accuracy. Our approach deals with applying a different weight for each network. Networks that had a lower classification error in the training set will have a larger weight when combining the results for each image from the test set. The algorithm for our proposed method can be seen in Algorithm 6.1.

Let \mathcal{R}^n be the n -dimensional feature space. Suppose, $\mathcal{X} = [x_1, x_2, \dots, x_n]^T$ be the n -dimensional feature vector, $\mathcal{X} \in \mathcal{R}^n$, $\Omega = [\omega_1, \omega_2, \dots, \omega_m]$ be the set of potential class labels and $C = [C_1, C_2, \dots, C_l]$ be the set of trained models for decision fusion.

Given the input pattern \mathcal{X} , the output of the i^{th} model is denoted as-

$$C_i(\mathcal{X}) = [P_{i,1}(\mathcal{X}), P_{i,2}(\mathcal{X}), \dots, P_{i,m}(\mathcal{X})]^T \quad (6.6)$$

where $C_{i,j}(\mathcal{X})$, $i = 1, 2, \dots, l$, $j = 1, 2, \dots, m$ represents the probability that \mathcal{X} belongs to class ω_j . Basically, $C_i(\mathcal{X})$ denotes the probability vector of model C_i . In our case $l = 4$ as there are 4 models and $m = 4$ as there are four classes. The fused output of l models is constructed as in:

$$C(\mathcal{X}) = F [C_1(\mathcal{X}), C_2(\mathcal{X}), \dots, C_l(\mathcal{X})] \quad (6.7)$$

where F is the fusion rule described below. Given some input data, the output probability vectors from all CNNs and one RFC are multiplied by a weight α before the prediction. So, for a given input \mathcal{X} , the output probability vector $C(\mathcal{X})$ is given by:

$$C(\mathcal{X}) = \sum_{j=1}^l \alpha_j C_j(\mathcal{X}) \quad (6.8)$$

where $C_j(\mathcal{X})$ is the output of the network j for a given input \mathcal{X} .

The weight α_j is primarily chosen by rank of the trained models. It is based on the order of accuracy in the validation set (the test fold in case of K-Fold cross validation) but the relative differences in accuracy of each model are taken into account. Let $R()$ be the function that gives the position of the model based on validation accuracy, sorted in increasing order. For example, the model with the largest accuracy will have a $R()$ value of m , where m is the number of classes ($m = 4$ in our case). The model with the second largest accuracy will have $R()$ a value of, $(m - 1)$ and so on until the model with the lowest accuracy will have $R() = 1$. Let, d_1, d_2, \dots, d_{l-1} be the differences in validation accuracies of l models which are themselves sorted in ascending order of validation accuracy. In our case, we have four models. Let their validation accuracies, after being trained on the full dataset,

Algorithm 6.1: MRA(*Accuracy*)

```

Input: Evaluation Accuracy of each model as  $Accuracy[A_1, A_2, \dots, A_l]$ 
Output: Weight for each model as  $Weights[\alpha_1, \alpha_2, \dots, \alpha_l]$ 
/* Store the original accuracies in  $AccRanks$  so that values in
Weights can be in the same order as Accuracy */
1  $AccRanks = Accuracy$ 
2  $sum_{factors} = l$ 
  //  $l = 4$  in our case
3 Sort the Accuracy array
4  $rank \leftarrow l$ 
5 foreach  $acc$  in  $Accuracy$  do           // Iterate through the sorted
  accuracies
6   for  $i \leftarrow 1$  to  $l$  do
7     if  $acc = AccRanks[i]$  then         /* If accuracy value equals  $i^{th}$ 
      index, then store the rank at the  $i^{th}$  index */
8        $AccRanks[i] \leftarrow rank$ 
9        $rank = rank - 1$ 
10    end
11  end
12 end
13 for  $j \leftarrow 1$  to  $l - 1$  do
14    $d_j \leftarrow Accuracy[l - j + 1] - Accuracy[l - j]$ 
15    $f_j = l - j + (1 - d_j)$ 
16    $sum_{factors} = sum_{factors} + f_j$ 
17 end
18  $Weights[1] \leftarrow \frac{l}{sum_{factors}}$ 
19 for  $i \leftarrow 2$  to  $l$  do
20    $Weights[i] \leftarrow \frac{f_{i-1}}{sum_{factors}}$ 
21 end
22 return  $Weights$ 

```

be denoted by A_1, A_2, A_3 and A_4 . The j^{th} difference d_j can be written as-

$$d_j = A_j - A_{j+1} \quad (6.9)$$

where $j = 1, 2, \dots, l - 1$. The main reason for calculating the differences in accuracy of the model is because the differences among the four models (in terms of accuracy) is not uniform. Hence, the differences in weight values for each model should also not be uniform. But, in normal rank based weighting method, each model gets a weight that is just 1 less than the previous model's weight in spite of having

accuracy values which are non-uniformly different. Thus, we decide to factor in the individual contributions of each model and penalize them according to the difference in accuracy. Based on this value of d_j , we calculate for each rank value, a factor f_j which is the sum of the rank value $R(A_j)$ and 1 minus the difference d_j . We write,

$$f_j = R(A_j) + (1 - d_j) \quad (6.10)$$

where $j = 1, 2, \dots, l - 1$. Finally, we calculate the weight α_j by normalizing the factor f_j :

$$\alpha_j = \frac{f_j}{\sum_{j=1}^{l-1} f_j + R_{max}} \quad (6.11)$$

where, $j = 1, 2, \dots, l - 1$ and R_{max} being the rank of the model with the highest accuracy ($R_{max} = 4$ in our case). The weights are then multiplied with the outputs of each model and hence Equation 6.8 can be represented as:

$$C(\mathcal{X}) = \sum_{j=1}^{l-1} \alpha_j C_j(\mathcal{X}) + \alpha_L C_L(\mathcal{X}) \quad (6.12)$$

In the above equation, $\alpha_L = \frac{R_{max}}{\sum_{j=1}^{l-1} f_j + R_{max}}$ with L denoting the model having the highest rank after sorting. The overall solution algorithm is presented in Algorithm 6.2.

6.6 Experimental Results

In this section, we first discuss data preparation. This is followed by implementation details. We then extensively evaluate our solution, including ablation studies and comparisons with external approaches.

6.6.1 Data Preparation

For experimentation, publicly available ADNI database is used [251]. We take a subset of ADNI DTI data that has been pre-processed with alignment and skull-stripping. Since there are patients that have multiple images taken during a period

Algorithm 6.2: Algorithm of proposed solution

Input : EPI volumes $E = [e_1, e_2, \dots, e_n]$
FA volumes $F = [fa_1, fa_2, \dots, fa_n]$
MD volumes $M = [md_1, md_2, \dots, md_n]$
FA-MD regional averages for 57 brain regions
 $V = [v_1, v_2, \dots, v_n]$ where each $v_i = [p_1, p_2, \dots, p_{57}, q_1, q_2, \dots, q_{57}]$,
 p_j being the averaged FA value and q_j being the averaged MD value for a
particular brain region.
 $n =$ number of training examples

Output: Classification label C

```

/* Initialize an Accuracy array of size 4 to store the
   accuracies of each model */
1 Accuracy = empty
/* Train EPI data in VoxCNN network and store accuracy */
2  $A_e \leftarrow \text{VoxCNN}_e(E)$ 
/* Train FA data in VoxCNN network and store accuracy */
3  $A_{fa} \leftarrow \text{VoxCNN}_{fa}(F)$ 
/* Train EPI data in VoxCNN network and store accuracy */
4  $A_{md} \leftarrow \text{VoxCNN}_{md}(M)$ 
/* Train EPI data in VoxCNN network and store accuracy */
5  $A_v \leftarrow \text{RFC}(V)$ 
6 Accuracy  $\leftarrow [A_e, A_{fa}, A_{md}, A_v]$ 
// Get the weights from Algorithm 6.1 by giving Accuracy as
   input
7 Weights  $\leftarrow \text{MRA}(\text{Accuracy})$ 
/* For any new input  $\mathcal{X} = [e, fa, md, v]$ , where  $e =$  EPI Volume,
    $fa =$  FA Volume,  $md =$  MD volume and  $v =$  Averaged FA-MD
   values, find the prediction results from each of the four
   models */
8  $C_e(e) \leftarrow \text{VoxCNN}_e(e)$ 
9  $C_{fa}(fa) \leftarrow \text{VoxCNN}_{fa}(fa)$ 
10  $C_{md}(md) \leftarrow \text{VoxCNN}_{md}(md)$ 
11  $C_{rfc}(v) \leftarrow \text{RFC}(v)$ 
12  $C(\mathcal{X}) \leftarrow [(Weights[0] \times C_e(e)) + (Weights[1] \times C_{fa}(fa)) + (Weights[2] \times$ 
    $C_{md}(md)) + (Weights[3] \times C_{rfc}(v))]$ 
13  $C \leftarrow$  index of  $\max(C(\mathcal{X}))$ 

```

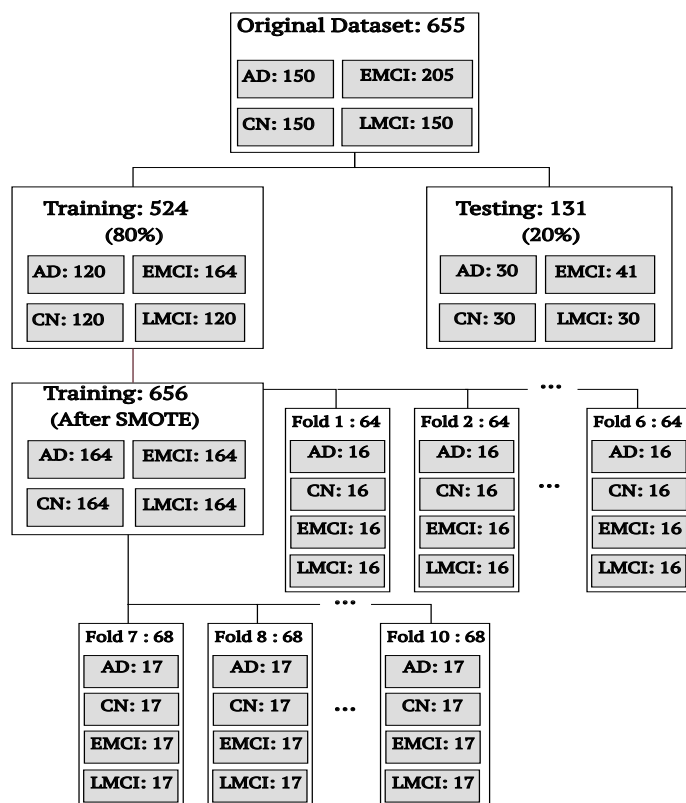


Figure 6.5: Data set division strategy

of time, to minimize possible information “leaks”, only the last images were taken for each subject. Also, there were data from two other classes namely MCI and Significant Memory Concern (SMC), but the number of data for these classes were so less that we did not consider those two classes for our experiments. The resulting dataset has 655 images of four classes: 150 of AD patients, 150 of LMCI, 205 of EMCI and 150 of CN. As the number of data for each class is unequal and the total number of data is not enough for training a deep learning model, SMOTE oversampling is used to increase the number of samples for each minority class. Each of the scan contains 3 3D volumes of size $110 \times 110 \times 110$ containing EPI, FA and MD data.

We first divided the dataset containing 655 images into training and testing sets, keeping in mind the number of images in each class. So we randomly took 80% of images each from AD, CN, EMCI and LMCI and created the training set containing total of 524 images. The remaining 20% from each class were taken to form the testing set, totaling 131 images.

We then applied SMOTE oversampling technique to increase the number of sam-

ples of each minority class (AD, EMCI and LMCI) to match that of the majority class (CN). As a result, each class has a total of 164 images now.

We then split the entire training set into 10 folds for 10-fold cross validation, keeping the number of images from each class the same. This resulted in 6 folds each containing 64 images (16 from each of AD, CN, EMCI and LMCI) and 4 folds each containing 68 images (17 from each of AD, CN, EMCI, and LMCI). The details of dataset division are explained in Fig. 6.5.

Each of the FA and MD data has a corresponding metadata file containing the averaged FA, MD values for each brain region which is extracted to create the 4th dataset containing real numbered values which was fed into RFC after going through the same dataset division process explained above. Each image file is stored as 3D tensor of shape $110 \times 110 \times 110$ in Nifti file format (.nii). The EPI image file contains voxel intensity values, and the FA and MD files have real numbered values within 0 and 1. For each model, a 10-fold cross validation has been employed to get better approximation of the prediction performance before applying the decision level fusion.

As such, the investigators within the ADNI contributed to the design and implementation of ADNI and/or provided data but did not participate in analysis or writing of this article. A complete listing of ADNI investigators can be found at the ADNI website [251].

6.6.2 Implementation Details

After creating training and testing sets for each type of data i.e., EPI, FA, MD and regional FA-MD values as explained in Section 6.6.1, we trained the training set with 10-fold cross validation. It performed better than 5-fold cross validation, as the number of training data in each fold significantly increased. We fed the accuracy value of each model (three CNNs for EPI, FA, MD and one RFC for FA-MD averaged values) to compute the modulated weights which were used in our novel modulated rank averaging method. Then, for each data in the testing set, the classification output of the individual models was multiplied with the corresponding

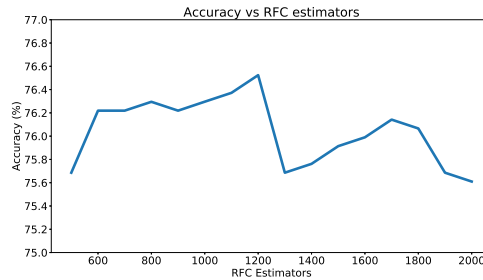


Figure 6.6: Graph showing how number of estimators in Random Forest Classification affects accuracy

weights previously calculated and a weighted averaging was done to arrive at the final classification.

We first furnish the details of VoxCNN training parameters. This is followed by elaborate discussions on setting of various parameters pertaining to RFC.

(A) VoxCNN Training parameters

We train the network using AdaM optimizer with learning rate of 27×10^{-6} and batch size of 32 for 80 epochs for each fold to get the perfect class separation on the fold training set and stabilize the performance metrics on fold validation set. The final accuracy is calculated on the separately kept test set containing 131 samples. We keep the training parameters the same for each model (EPI, FA, and MD) in order to avoid conflict while combining the outputs of the models so that the decision level fusion is unbiased.

(B) Random forest classifier parameters

The classifier is also tuned on the training set by experimenting with the number of trees/estimators. From Fig. 6.6, it is evident that setting the number of trees to 1200 yields best results. The function to measure split quality is set to ‘entropy’ as ‘gini’ function often leads to poorer results than ‘entropy’.

(C) Synthetic minority oversampling technique parameters

SMOTE oversampling (see Section 6.5.2) is used to create synthetic training samples, which helps in balancing the number of training data for each class. After

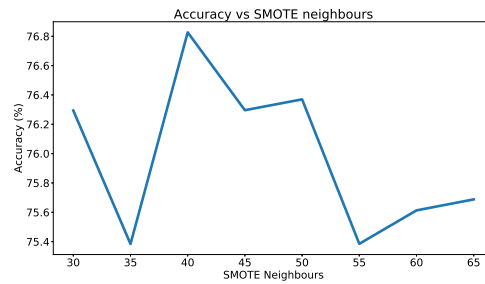


Figure 6.7: Graph showing how number of SMOTE neighbors affects accuracy

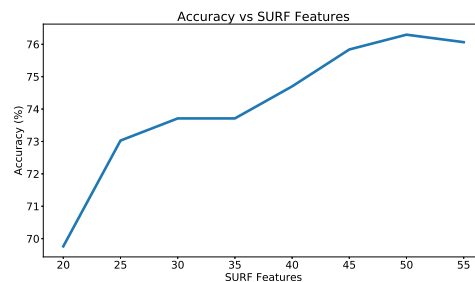


Figure 6.8: Graph showing how number of SURF features affects accuracy

experimenting with the number of neighbors in SMOTE on the training set, keeping other parameters constant, we have found that the best results are obtained when the number of neighbors is set to 40. This is evident in the Accuracy vs. SMOTE neighbors graph in Fig. 6.7.

(D) Spatially Uniform Relief parameters

SURF selects the best features from numerous features to find a balance between computation time and accuracy. In our case, we have selected 50 features. By experimenting with the number of features in a small dataset keeping the number of RFC estimators and SMOTE neighbors constant, it was found that 50 features gives the best accuracy value as evident in Fig. 6.8.

6.6.3 Time Complexity Analysis

Our entire solution constitutes three main components - 3D VoxCNN model for using 3D radiological data, Random Forest Classifier for using FA-MD values and MRA algorithm for ranking the mentioned models. Using the time complexity (T_V^{3D})

of 3D VoxCNN previously defined in Eq. 4.12 of Chapter 4, and that of Random Forest (T_{rfc}) from Eq. 2.10 of Chapter 2, we now define the time complexity of MRA algorithm below-

From Algorithm 6.1, it follows that the time complexity of the algorithm depends on the number of models m_n used. In this case, $m_n = 4$. Hence, the time complexity can be written as-

$$T_{MRA} = O(m_n) \quad (6.13)$$

Thus, the total time complexity of our proposed solution is-

$$\begin{aligned} T_{total} &= T_V^{3D} + T_{rfc} + T_{MRA} \\ &= \sum_{n=1}^d k_{n-1} \cdot s_n^3 \cdot f_n \cdot l_n^3 \cdot r_1 \cdot b_1 + \sum_{l=1}^f D \cdot W \cdot H \cdot N + O(R \cdot p \cdot q \cdot \log(p)) + O(m_n) \end{aligned} \quad (6.14)$$

In this case, $d = 4$ and $f = 1$, i.e., we have 4 convolutional layers and 1 fully connected layer in VoxCNN.

6.6.4 Individual Model Performance

The evaluation accuracy and Area under curve (AUC) of Receiver Operating Characteristics (ROC) for the individual models evaluated with the test set are shown in Table 6.1. The ROC is plotted by varying the decision threshold of our model output and then plotting the True Positive Rate (TPR) vs. False Positive Rate (FPR) at each threshold. It can be seen that the RFC has performed worst, while the EPI VoxCNN model has performed best. The models in decreasing order of performance are EPI, MD, FA and RFC.

6.6.5 Combined Model Performance

The models in various combinations are fused at decision level using three different fusion techniques as mentioned below.

Table 6.1: Individual evaluation accuracy of EPI, FA, MD and RFC models

Model	Accuracy (%)
EPI	87.4
FA	76.4
MD	84.5
RFC	69.4

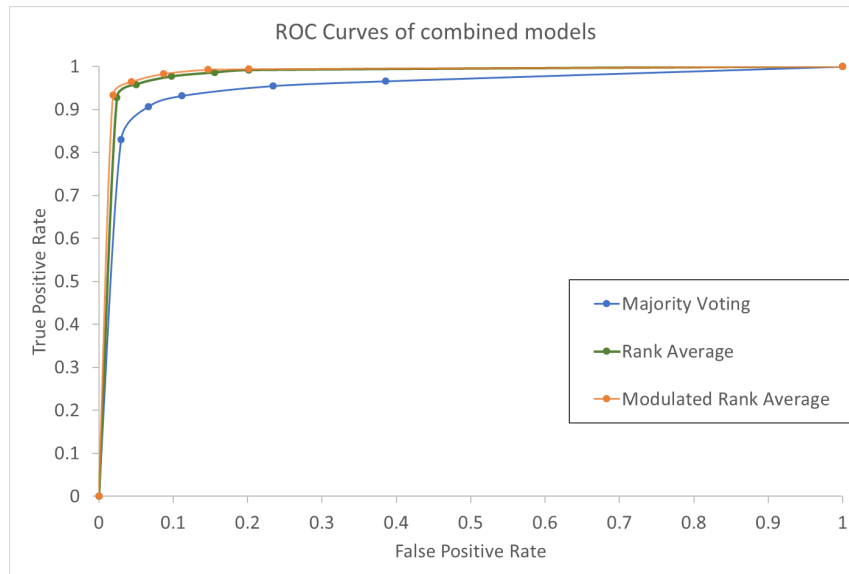


Figure 6.9: ROC curves of the combined models (Majority Voting, Rank Averaging and Modulated Rank Averaging)

(A) Majority voting method

Since each model gives a single class decision, we take that as a vote for that particular class and consider the final output as the class that gets the maximum number of votes. In this way, we found that the accuracy has significantly increased from that of the individual models.

(B) Rank Averaging method

In this scheme, the output probabilities of each model are simply multiplied by the rank of that model, sorted according to descending accuracy [252]. This method is better than the majority voting method by 3%.

(C) Modulated Rank Averaging method

Our proposed model performed even better than the Rank Averaging method. As the accuracy of each model sorted in increasing order is not uniform (see Table 6.1), i.e., their relative differences are not the same, the weight assigned to them should also not be uniform. We modified the weights of the Rank Averaging method to factor in the differences of the model accuracies (see Eq. 6.10). The ROC curves for each of the combined models are plotted in Fig. 6.9 where we can see that the modulated rank average performs slightly better than the rank averaging model. This better performance is quantitatively corroborated by the accuracy and AUC values in Table 6.3. The per-class metrics are also shown in Table 6.4.

To demonstrate statistical significance of the improvement in results, we have computed confidence interval (with a confidence of 95%) for the difference between evaluation accuracy values of the proposed modulated rank averaging method and other two decision fusion approaches, i.e., rank averaging and majority voting. If the confidence interval includes zero, the difference is not significant at that confidence level. If the confidence interval does not include zero, then the sign of the differences in the accuracy values indicates which alternative is better [253]. Since, the confidence intervals (with a confidence of 95%) do not include zero in either of the cases, we can say that the results presented in Table 6.2 confirm that the proposed decision level fusion approach yields statistically significant improvements over the other two existing decision fusion strategies.

6.6.6 Ablation Study of the Models

Keeping the best fusion strategy, i.e., modulated rank averaging based decision fusion, we now show the utility of the four models. An extensive ablation study is

Table 6.2: Differences in evaluation accuracy at 95% confidence level

Difference	Confidence Interval (95%)	
	Min.	Max.
Ours vs. Rank Averaging	3.298×10^{-4}	167.854×10^{-4}
Ours vs. Majority Voting	294.871×10^{-4}	528.495×10^{-4}

Table 6.3: Evaluation accuracy and Area under Curve (AUC) of the fusion methods with all four model combinations

Fusion Approach	Accuracy (%)	AUC
Majority Voting	88.7	0.923
Rank Averaging	91.8	0.957
Modulated Rank Averaging	92.6	0.962

Table 6.4: Per class metrics containing the precision, recall and f1-score for each disease class

Class	Precision	Recall	f1-score
AD	0.96	0.92	0.94
CN	1.00	0.97	0.98
EMCI	0.88	0.88	0.88
LMCI	0.85	0.92	0.88

carried out in that regard. Within that study, we compare the performance with all 4 models (EPI, FA, MD and RFC), all possible combinations of any 3 models at a time as well as any 2 models at a time. The accuracy and AUC for each combination are shown in Table 6.5. This table shows that all four models are necessary to yield best result. It is also interesting to note from a comparison of Table 6.1 and Table 6.5 that inclusion of RFC, a classical ML tool, can improve the overall accuracy when combined with DL models on EPI or MD values. Furthermore, when clubbed with FA + MD, RFC can once again increase the overall accuracy. So, we show how effective can be the combination of DL and ML methods for a given problem.

6.6.7 Comparison with feature fusion strategies

We also ran an experiment to apply feature fusion instead of decision level fusion. Features were extracted from the individual DL models (EPI, FA and MD), from the penultimate Fully Connected (FC) layer (dense_1). It consisted of 64 dimensions for each of the EPI, FA and MD models. Concatenating all the features from the DL models along with the input features of the RFC model (6.5.2), a total of 242 features were obtained (64 + 64 + 64 + 50). All these features were then trained in an RFC with 1200 trees. This approach of classification gave a mean accuracy of 79.34%

Table 6.5: Accuracy and Area Under Curve (AUC) for all combinations using Modulated Rank Averaging method

Combinations	Accuracy (%)	AUC
All	92.6	0.962
EPI + FA + MD	91.2	0.955
EPI + FA + RFC	86.48	0.913
EPI + MD + RFC	88.87	0.934
FA + MD + RFC	85.32	0.908
EPI + FA	85.26	0.901
EPI + MD	90.42	0.943
EPI + RFC	84.3	0.886
FA + MD	89.37	0.928
FA + RFC	72.74	0.815
MD + RFC	85.29	0.894

Table 6.6: Comparison of evaluation accuracy with other state-of-the-art approaches

Approach	Modalities	Number of classes	Accuracy (%)
Bi <i>et. al.</i> (2019) [254]	MRI	3	92.5
Billones <i>et. al.</i> (2016) [22]	MRI	3	91.85
Vu <i>et. al.</i> (2018) [255]	MRI + PET	3	91.13
Cheng <i>et. al.</i> (2017) [256]	MRI	3	87.15
Duc <i>et. al.</i> (2020) [224]	MRI	3	87.15
Gunawardena <i>et. al.</i> (2017) [257]	MRI	3	84.4
Our Method	DTI	4	92.6

after doing 10-fold cross-validation. We also compare our proposed modulated rank averaging method with some state-of-the-art feature fusion approaches in Table 6.7. All the above approaches perform either a binary or a three-way classification. It is interesting to note that our proposed approach has achieved 92.6% accuracy in a direct 4-class classification problem. Hence, we confirm that for this classification problem, decision level fusion works better than feature fusion.

6.6.8 Comparison with other approaches

We compare our method with seven state-of-the-art methods but no results are available for 4-class AD classification on ADNI dataset using DTI data. So, we are showing comparisons from the problem perspective and comparing our results

Table 6.7: Comparison of evaluation accuracy among different feature fusion methods along with the accuracy (in %) for five types of classifications (AD vs. CN, AD vs. MCI, MCI vs. CN, AD vs. MCI vs. CN, AD vs. EMCI vs. LMCI vs. CN)

Approach	Accuracy (%) with Classes				
	AD vs. CN	MCI vs. CN	AD vs. MCI	AD vs. CN vs. MCI	AD vs. CN vs. EMCI vs. LMCI
Li et. al. (2015) [258]	91.4	82.1			
Shi et. al. (2017) [259]	91.95	83.72			
Lei et. al. (2016) [260]	96.93	82.75			
Madusanka et. al. (2019) [261]	86.61	82.05	78.95		
Xiao et. al. (2017) [262]	85.71	86.11	79.44	75	
Our feature fusion method					79.34
Our decision fusion method					92.6

with the works which have addressed at the same AD classification problem but from differing modalities and fewer classes. All seven methods showed classification among AD, MCI and CN, i.e., three classes. But we have segregated the MCI class into more detailed EMCI and LMCI classes and thus made a four class classification. Classification between EMCI and LMCI is particularly difficult as their differences are not very significant. Also, none of them used DTI modality. They used MRI or a combination of MRI and PET to achieve their goal. The comparisons, shown in Table 6.6, clearly points to the supremacy of our approach.

6.7 Summary

Classification of Alzheimer’s Disease (AD) is crucial in diagnosing dementia, particularly in the aging population. Currently, neurologists primarily perform this classification manually using brain scans. Existing automated methods are limited to two-class or three-class classification from MRI data. In this chapter, we present the first automated solution for four-class AD classification using 3D DTI data. We trained deep learning (VoxCNN) and machine learning (Random Forest) models on different aspects of the DTI scans, and combined their outputs using a modulated rank averaging decision fusion strategy. Extensive experiments, including comparisons and ablation studies on the ADNI database, demonstrate the effectiveness of our approach.

Chapter 7

Segmentation of the Hippocampus of the human brain

The development of computer vision techniques for hippocampus segmentation is crucial for accurately quantifying hippocampal volume, a key biomarker in diagnosing and monitoring neurodegenerative diseases like Alzheimer's and epilepsy. Precise segmentation enables early detection of structural changes, facilitating timely interventions that can slow disease progression and improve patient outcomes. These techniques also reduce errors associated with manual segmentation, enhancing diagnostic accuracy and the effectiveness of treatment monitoring.

7.1 Introduction

The hippocampus (HC) is a bilateral brain structure located in the medial temporal lobe at both sides of the brainstem near the cerebellum. HC is a very intricate and heterogeneous structure broadly divided into several subfields, namely, subiculum, cornu ammonis (CA1/2/3/4), and dentate gyrus (DG). Alzheimer's Disease (AD), Parkinson's Disease (PD), autism, multiple sclerosis, and natural ageing affect the HC. Stages of AD affect hippocampal subfields differently. The atrophy of stratum radiatum/stratum lacunosum-moleculare (SRLM) apical dendrites of hippocampal CA1 closely mirrors episodic memory impairment in AD as well as PD patients [263]. HC atrophy study may give biomarkers and improved techniques for detecting and predicting AD and other neurodegenerative illnesses. So, segmenta-

tion of HC assumes paramount importance in neuroradiology. HC is involved in many brain functions such as memory and spatial reasoning, [264], but can be affected in shape and volume by different pathologies, such as the neurodegeneration associated with Alzheimer’s Disease (AD) [265], or surgical intervention to treat temporal lobe epilepsy [266]. The volume and shape changes of hippocampus serve as early indicators of tissue degeneration. Therefore, intercepting these changes is of crucial importance. Over the past decade, there has been a growing interest in segmenting hippocampal subfields using MRI. Recent research has identified distinct functional roles for these anatomical subregions, with CA1 implicated in memory integration and inference [267], CA3 in memory retrieval [268], and both the dentate gyrus (DG) and CA3 in pattern separation [269]. Clinically, the volume or morphology of the hippocampus and its subfields are closely related to many neurodegenerative diseases like Epilepsy [270] and Alzheimer’s disease [271]. So, it is desirable to develop automatic hippocampal subfields segmentation from brain MR image. However, manual delineation of hippocampal subfields is a laborious and time-intensive task, leading to constraints on sample sizes in various studies.

7.2 Related Works

This high expense of manual segmentation has prompted the development of efficient automation techniques. We first discuss some works that use classical techniques for HC subfield segmentation. FreeSurfer (FSL) [272] was one of the early attempts. FSL is capable of producing decent coarse segmentation. However, it lacks finer details and accurate edge delineation. Some other noteworthy works attempted HC segmentation using atlas registration [273] and level sets [274], but they were in 2D and did not perform well in 3D. In order to perform fine-grained multi-class segmentation, several methods were proposed. The authors in [275] used a multi-atlas approach combined with a similarity-weighted voting and a boosting-based error correction as a solution. They termed their method, ASHS. This method took several hours to produce a segmentation due to exhaustive use of non-linear registrations. More recently, a method named HIPS [276] obtained state-of-the-art results with

relatively low processing times. While classical methods integrated domain-specific image features like gradient, intensity, and textures within an energy minimization framework, they are found to heavily depend on initialization, such as manual seeding. As a result, they are prone to segmentation errors caused by uncertain positioning of the markers. Furthermore, these approaches are quite laborious and may not be practical for clinical environments with a heavy workload.

In recent times, several deep learning (DL) techniques have been used to improve HC segmentation. In recent times, Convolutional Neural Networks (CNN) gained popularity over previous semi-automatic and automated approaches for HC segmentation [277]. Medical imaging lacks consistent and sufficient annotated data, making DL algorithms performing sometimes below par [278]. To circumvent this limitation, shape-driven DL algorithms have evolved, which can combine CNN-learned deep features with structural shape information, to boost the performance [279]. Only a few shape-based HC segmentation studies are reported till date. Brusini *et al.* [280] employed shape fitting and UNet to segment the HC. However, they used three 2D models instead of one 3D model, which requires more training time. Others like Tang *et al.* [281] have employed signed distance maps (SDM) to build loss functions that penalize Dice loss and SDM values. The above technique estimated loss by training a model to predict SDM and calculating Dice loss between the predicted and the ground truth SDM. This required additional model training. In the scenario of multi-class segmentation of HC subfields, many DL techniques have been proposed. UNet based methods [282], [283] have shown promising results. Shi *et al.* [284] proposed a Generative Adversarial Network (GAN) to create a segmentation model. But UNet and GAN based methods require a lot of data and medical imaging lacks consistent and sufficient annotated data, making DL algorithms perform poorly in many cases [285]. Although some authors have tried to bypass this problem using multi scaling technique [286] and using higher resolution data [287], the problem continues to exist.

7.3 Contributions

In this chapter, we discuss two major tasks related to HC segmentation. The first task is to segment the whole HC and the second task is to segment the subfields of HC. We discuss the contributions related to both these tasks below-

7.3.1 Whole Hippocampus Segmentation

In this chapter, we first propose a novel shape-based DL strategy for HC segmentation. Contour directional information is incorporated to create a shape based model using Histogram of Oriented Gradients (HoG). HoG [288] is an excellent feature descriptor that can extract shape as well as structural information from an image by computing magnitude and angle gradients. The application of using HoG with deep learning is well demonstrated in [289]. We took inspiration from [290] while developing the formalism. We have further suggested an efficient pooling to tackle the derivability problem of histogram in HoG. Normal pooling methods like max-pool, min-pool and average-pool may not yield desirable results in this situation. Our main contributions are summarized below -

1. We have incorporated shape based loss using 3D gradient magnitude and gradient angles of HoG to find the 3D HC boundary along with directional information.
2. A 3D weighted majority pooling technique, that can act as a substitute for histogram operation on tensor data, is developed. This can be implemented in DL models as it is derivable.
3. Our proposed solution for HC segmentation can provide highly accurate results in 3D. This can reduce the diagnosis time and expedite the surgical planning.

7.3.2 Hippocampus Subfields Segmentation

In order to segment subfields, we further propose a shape driven multi-class segmentation method using UNet [291] and graph cut. We take inspirations from [83], [87] and [292] to create a state-of-the-art model to segment the HC into three classes,

namely, CA1-3, CA4/DG and Subiculum (Sub). In [118], Boykov *et al.* showed that a two-class segmentation is achievable in polynomial time using graph cuts. However, if the number of labels exceeds 2 (as for the present problem), finding an exact solution becomes an NP-hard problem. They suggested two types of large moves (changing labels of individual pixels/voxels) based on minimal graph cuts, namely, α -expansion and α - β swap. In this work, we use UNet to improve the α - β swap. A number of research works demonstrated that use of shape information can improve the segmentation accuracy [293], [294], [87]. Shape priors provide valuable guidance by incorporating prior knowledge about the expected shapes of objects in the image. This guidance helps the segmentation algorithm to make more informed decisions about the boundaries and regions of interest [295]. By imposing shape constraints, shape priors help to enforce consistency in the segmented shapes, ensuring that the output conforms to the expected shape characteristics [296]. The shape prior has to be made adaptive in case of substantial noise and intensity variations. Here, we better an adaptive shape prior from deep learned information via UNet. Our main contributions are summarized below -

1. We propose a new energy function for multi-class segmentation based on graph cut and deep learning (3D UNet).
2. We incorporated the learned information from 3D UNet for optimizing the number of α - β swaps used in multi-class graph cut segmentation.
3. In addition, we also show how an adaptive shape prior can be learned from UNet thereby improving the overall segmentation performance.

7.4 Proposed Method for Whole Hippocampus Segmentation

In this section, we describe our proposed method. The block diagram showing the overall method is presented in Fig. 7.1.

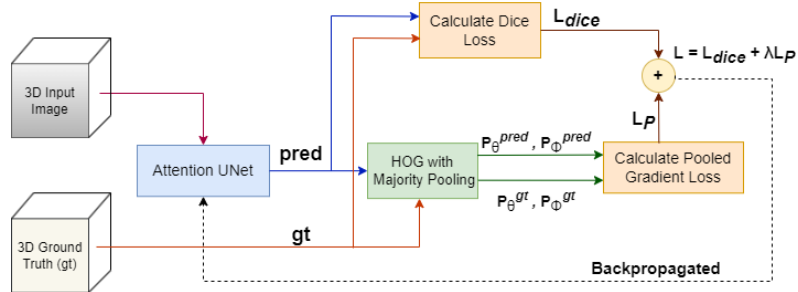


Figure 7.1: Block diagram of our proposed approach: $pred$ denotes the predicted output of Attention Unet model and gt denotes the ground truth. P_{pred} and P_{gt} are the pooled gradients of predicted and ground truth respectively, as shown in Eq. 7.9.

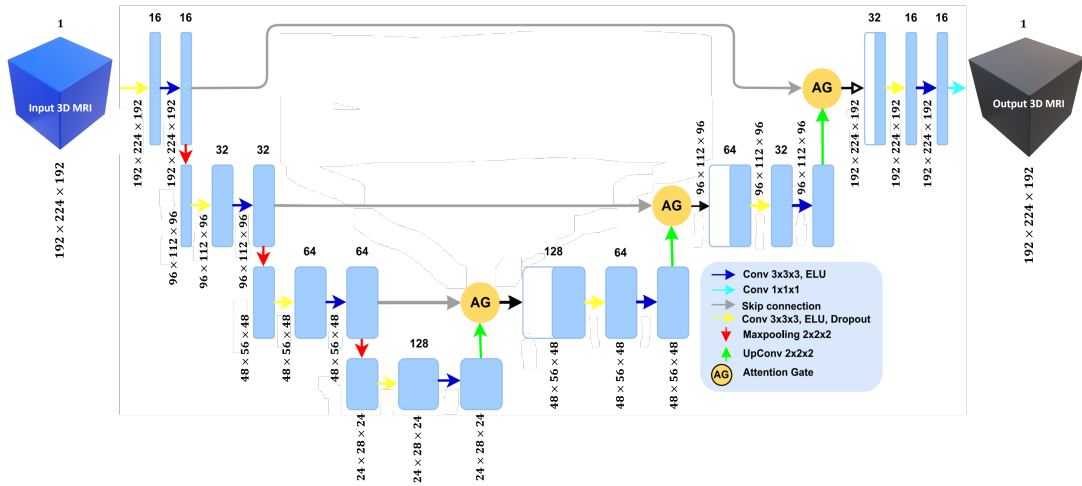


Figure 7.2: Architecture of 3D Attention UNet. The input, output and feature dimensions are written beside the corresponding blocks. The number of channels is written on top of each block.

7.4.1 Attention UNet

3D Unet [33], a deep fully convolutional encoder-decoder network architecture has gained popularity for voxel-wise segmentation. In this work, we utilise the Attention based UNet as described in [297]. The authors in [298] have argued that it is one of the best Unet models in terms of accuracy, number of parameters, training and inference time. The attention mechanism is necessary for training our model with a focus on the HC region. This focus has to be explicitly induced due to the inherent difficulty in distinguishing HC from its neighboring tissues, which exhibit similar contrast. The Attention UNet consists of a contracting path, which incorporates convolutional and max-pooling layers, followed by an expanding path comprising convolutional and up-sampling layers. The key distinction between the Attention UNet and the standard UNet lies in the integration of attention mechanisms, or attention gates, within the skip connections. These gates enhance the network's ability to focus on relevant features while suppressing irrelevant ones, thereby improving the accuracy of tumor segmentation. As explained in [297], the attention gate (AG) takes two input vectors: Q and S . Vector S is obtained from a deeper layer of the network, offering lower spatial resolution but richer feature representation, while vector Q undergoes a strided convolution, and vector S is processed through a 1×1 convolution. The vectors are then combined through element-wise summation to form a new vector, which is passed through activation and convolutional layers for dimensionality reduction. A sigmoid function is applied to generate attention coefficients, representing the importance of each element in vector Q . Trilinear interpolation [299] is used to restore the attention coefficients to their original dimensions. These coefficients are multiplied with vector Q , scaling it according to its relevance, before the modified vector is propagated through the skip connection for further processing. Fig. 7.2 depicts the architecture of our 3D Attention UNet model. It has a depth of 4 and takes as input a 3D single channel tensor of dimension $192 \times 224 \times 192 \times 1$ (*height* \times *width* \times *depth* \times *channel*). The model outputs a 3D single channel tensor of same dimensions containing the segmentation mask. We use batch normalization for each convolutional layer followed by ELU activation.

7.4.2 Basics of 2D HoG

HoG is a well-known feature descriptor which is able to extract structural and shape information [288]. It is better than other feature descriptors as it uses magnitude as well as angle of the gradient to compute the features. But the HoG descriptor focuses on the structure or the shape of an object. It is better than other edge descriptor as it uses magnitude as well as angle of the gradient to compute the features. For the regions of the image, it generates histograms using the magnitude and orientations of the gradient.

Considering a block of 3×3 pixels, first G_x and G_y is calculated for each pixel using the horizontal and vertical 1-D filters $[1, 0, -1]$ and $[1, 0, -1]$. G_x and G_y is calculated using the formulae below for each pixel value.

$$\begin{aligned} G_x(r, c) &= I(r, c - 1) - I(r, c + 1) \\ G_y(r, c) &= I(r - 1, c) - I(r + 1, c) \end{aligned} \tag{7.1}$$

where (r, c) corresponds to row and column number and $I(x, y)$ means intensity of pixel at location (x, y) .

After calculating G_x and G_y , magnitude (μ) and angle (θ) of each pixel is calculated using-

$$\mu = \sqrt{G_x^2 + G_y^2} \tag{7.2}$$

and after obtaining the gradient of each pixel, the gradient matrices μ and θ are divided into $n \times n$ cells. If the height and width of a 2D image is h and w respectively, then each cell will contain $\frac{h}{n} \times \frac{w}{n}$ pixels. h and w should be divisible by n to avoid discrepancy and information loss. Hence, the image should be resized accordingly before applying HoG. For each cell, a histogram is calculated followed by block normalization as described in [288].

7.4.3 Basics of 3D HoG

Since, we deal with 3D volumetric images, we use 3D feature descriptors which are shown to carry more information than their 2D counterparts [300]. We first extract

gradients from an image volume A of dimension $h \times w \times d$; where h , w and d respectively denote the height, width and depth of A . We calculate G_x , G_y and G_z , the gradient matrices along the x , y and z dimensions respectively by convolving A with a block of $3 \times 3 \times 3$ kernel. The elements of the matrices are as follows:

$$\begin{aligned} G_x(i, j, k) &= I(i + 1, j, k) - I(i - 1, j, k) \\ G_y(i, j, k) &= I(i, j + 1, k) - I(i, j - 1, k) \\ G_z(i, j, k) &= I(i, j, k + 1) - I(i, j, k - 1) \end{aligned} \quad (7.3)$$

Here, i, j, k denote coordinates along the x, y, z directions. $I(i, j, k)$ is the intensity at $A(i, j, k)$. We build three matrices denoting the magnitude (V_μ), the azimuth angle (V_θ) and the zenith angle (V_ϕ) of the gradient. The elements of the matrices are respectively given by:

$$V_\mu(i, j, k) = \sqrt{G_x^2 + G_y^2 + G_z^2} \quad (7.4)$$

$$V_\theta(i, j, k) = \left| \tan^{-1} \frac{G_y}{G_x} \right| \quad (7.5)$$

$$V_\phi(i, j, k) = \left| \cos^{-1} \frac{G_z}{V_\mu} \right| \quad (7.6)$$

Naturally, each of V_μ, V_θ, V_ϕ has the dimension of A , *i.e.*, $(h \times w \times d)$.

7.4.4 Majority Pooling

The histogram, which is an integral part of HoG, is not a derivable function. This is a critical problem as for deep learning models, we need derivable functions to back propagate the loss. Hence, we present a novel majority pooling technique that pools the magnitude of the voxels with the most frequently occurring angle (separately in V_θ & V_ϕ) in each cell.

We first divide the three matrices V_μ, V_θ & V_ϕ into $n \times n \times n$ voxel grids (smaller

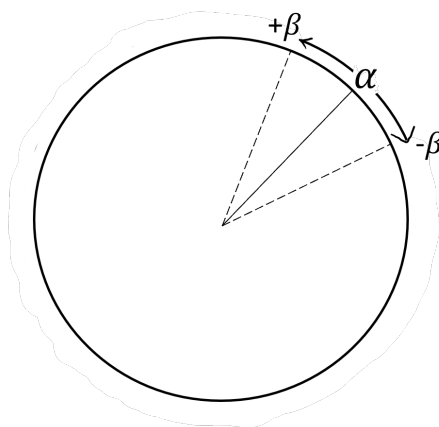


Figure 7.3: Figure showing the range of angles considered for selecting the voxels from v_θ^s and v_ϕ^s . If α is the most frequently occurring angle, then all angles in range $\alpha \pm \beta$ are considered.

matrices). So, each grid will contain $\frac{h}{n} \times \frac{w}{n} \times \frac{d}{n}$ voxels. Let us consider the total number of such grids as g . Then, we can write $V_\mu = [v_\mu^1, v_\mu^2, \dots, v_\mu^g]$ where the smaller matrices $v_\mu^1, v_\mu^2, \dots, v_\mu^g$ are constructed row-wise with a stride of n and each of them has dimensions $n \times n \times n$. Similarly, we write $V_\theta = [v_\theta^1, v_\theta^2, \dots, v_\theta^g]$ and $V_\phi = [v_\phi^1, v_\phi^2, \dots, v_\phi^g]$. We next construct two sets ψ_θ and ψ_ϕ to respectively select the voxels from v_θ^s and v_ϕ^s that have non-zero magnitudes in the corresponding location of v_μ^s ($s = 1, 2, \dots, g$). So, we write:

$$\begin{aligned} \psi_\theta &= \{v_\theta^s(i, j, k) | v_\mu^s(i, j, k) > 0\} \\ \psi_\phi &= \{v_\phi^s(i, j, k) | v_\mu^s(i, j, k) > 0\} \end{aligned} \quad (7.7)$$

where, $v_\theta^s(i, j, k)$, $v_\phi^s(i, j, k)$ and $v_\mu^s(i, j, k)$ is the value at location (i, j, k) of these matrices ($i = 1 \dots n, j = 1 \dots n, k = 1 \dots n$). We now find the most frequently occurring angle α from ψ_θ and save the voxel location of all the angles in the range $[\alpha - \beta, \alpha + \beta]$ (see Fig. 7.3) into M_θ , where $M_\theta \subset \psi_\theta$. This is done to pool magnitude values from v_μ^s with corresponding angles in v_θ^s that are equal to or close to α . This range is suitable as the gradient directions of all the angles within this range give a similar idea about the main direction of that region. We construct M_ϕ likewise, where, $M_\phi \subset \psi_\phi$.

Finally, we create matrices p_θ^s, p_ϕ^s to pool only those values from v_μ^s contained in

M_θ and M_ϕ respectively and set the values at other locations to zero. So, we write:

$$p_\theta^s(i, j, k) = \begin{cases} v_\mu^s(i, j, k) & \text{if } (i, j, k) \in M_\theta \\ 0 & \text{otherwise} \end{cases} \quad (7.8)$$

$$p_\phi^s(i, j, k) = \begin{cases} v_\mu^s(i, j, k) & \text{if } (i, j, k) \in M_\phi \\ 0 & \text{otherwise} \end{cases}$$

where $p_\theta^s(i, j, k)$ is the value at location (i, j, k) of p_θ^s and similarly for p_ϕ^s . All p_θ^s ($s = 1, \dots, g$) of dimension $n \times n \times n$ are concatenated row-wise to form the pooling matrix P_θ of dimension $h \times w \times d$. We construct P_ϕ likewise.

The above pooled matrices contain gradient magnitudes of only those regions that have voxels with dominating angles in that region. Hence, it represents an approximate surface of the object with angular dominance capturing the shape information.

7.4.5 A Shape based Loss Function

After formulating the 3D pooling matrices in the above section, it can be applied to both the ground truth label mask and the predicted label mask. Let us denote the output of 3D pooled gradient matrix from the ground truth mask as P_θ^{gt} , and P_ϕ^{gt} , where ϕ and θ denote the azimuth and zenith angles respectively. Similarly, the corresponding pooled gradient matrices of the predicted mask are denoted as P_θ^{pred} and P_ϕ^{pred} . Thus, the loss function can be calculated as Sum Squared Error (SSE) between the ground truth pooled gradient matrices and predicted pooled gradient matrices of azimuth (θ) and zenith (ϕ). We can represent the loss function based on the pooled gradient matrices as:

$$\mathcal{L}_P = \sqrt{[P_\theta^{gt} - P_\theta^{pred}]^2 + [P_\phi^{gt} - P_\phi^{pred}]^2}$$

$$= \sqrt{\sum_{i=1}^h \sum_{j=1}^w \sum_{k=1}^d \left[\left\{ p_\theta^{gt}(i, j, k) - p_\theta^{pred}(i, j, k) \right\}^2 + \left\{ p_\phi^{gt}(i, j, k) - p_\phi^{pred}(i, j, k) \right\}^2 \right]} \quad (7.9)$$

where $p_{\theta}^{gt}(i, j, k)$ is the value of P_{θ}^{gt} at position (i, j, k) . Similar meaning holds for $p_{\theta}^{pred}(i, j, k)$, $p_{\phi}^{gt}(i, j, k)$ and $p_{\phi}^{pred}(i, j, k)$. The above loss function focuses on the shape of the predicted mask and penalizes on the dissimilarity of the shape of the prediction. However, we also need to focus on the overall structure and region for correct prediction. So, we use the Dice Loss [126] along with our pooled gradient loss to build a composite function. This composite loss function is given by:

$$\mathcal{L} = \mathcal{L}_{dice} + \lambda \mathcal{L}_P \quad (7.10)$$

where λ denotes a weight with value in $(0, 1]$ and is adjusted experimentally.

7.5 Hippocampus Subfield Segmentation

In this section we discuss about how we modified Graph Cut with deep learned information to perform state of the multi-class segmentation of Hippocampus.

7.5.1 Deep Graph Cut

Let us define the 3D MRI input as a gray-scale volumetric data, which may be represented as a 3D weighted graph denoted by $G = G(V, E)$. Each vertex is represented by a voxel x in G , and X is the collection of all voxels. We introduce two new vertices, called 'source' and 'sink', represented by s and t respectively. There are two sorts of edges or linkages that we consider: t-links (T) and n-links (N). s and t is linked to every voxel x through t-links. We utilize a compact 26-neighborhood, represented as $Ne(x)$ for every voxel x . Assume that y is a neighbor of x . Therefore, y belongs to the neighborhood of x , and we establish a connection between x and y by an n-link. Therefore, the set V is defined as the union of sets X , s , and t , whereas the set E is defined as the union of sets T and N . Let us establish a segmentation A as a classification of all voxels into two distinct classes: "object" or "background". This classification is done on a voxel-wise basis. Therefore, according to the reference [301], it is necessary to minimize the subsequent energy function:

$$\zeta(A) = B(A) + \lambda R(A) \quad (7.11)$$

The term $B(A)$ represents the boundary characteristics or smoothness term of A , while $R(A)$ represents the regional properties or data term of A . These terms are represented mathematically as below:

$$B(A) = \sum_{x \in X, y \in Ne(x)} B_{(x,y)} \quad (7.12)$$

$$R(A) = \sum_{x \in X} R_x \quad (7.13)$$

In [83], we modified the above energy function (Eq. 7.11) by incorporating learned information from the 3D UNet [291]. The modified energy function is given below:

$$\begin{aligned} \zeta_{DGC}(A) = & \sum_{x \in X, y \in Ne(x)} B_{DGC}(x, y) + \\ & \lambda_{DGC}(x) \sum_{x \in X} R_{DGC}(x) \end{aligned} \quad (7.14)$$

7.5.2 Multi-class Deep Graph Cut

As stated earlier, in this work, we deal with multi-class hippocampus segmentation where a voxel x can belong to any of 'CA1-3', 'CA4/DG' and Subiculum. Following [118], our goal is to find a labeling f that assigns each voxel $x \in X$ a label $f_x \in \mathcal{L}$ and, $|\mathcal{L}| > 2$, where f is both piecewise smooth and consistent with the observed data. Any labeling f can be uniquely represented by a partition of image voxels, $V = \bigcup_{l \in \mathcal{L}} V_l$, where $V_l = \{x \in V | f_x = l\}$ is a subset of pixels assigned a label l . Hence, Eq. 7.14 can be rewritten as:

$$\zeta_{DGC}(A_f) = \sum_{x \in X, y \in Ne(x)} B_{DGC}(f_x, f_y) + \lambda_{DGC}(f_x) \sum_{x \in X} R_{DGC}(f_x) \quad (7.15)$$

where $\zeta_{DGC}(A_f)$ is the energy of the labelling f . The knowledge acquired from the 3D UNet [291] is included into the energy function of the 3D graph cut algorithm in order to achieve precise segmentation. The 3D probability map is obtained from the last convolutional layer for each image. This map is then used to determine the

probability, denoted as $Pr(f_x)_{UN}$, of any voxel x belonging to the label f_x . The 3D UNet calculates a regression function that maps the voxels of a 3D input to a 3D voxel-wise probability map. This is denoted as $\mathcal{P} : \mathbb{R}^3 \rightarrow (0, 1)$, and it assigns a value between 0 and 1 to each voxel. Additionally, this probability map is used as an automated seed required by 3D graph cut algorithm. With this, we now explain the smoothness and data term in the context of multi label problem as follows. As mentioned in [83], $B_{DGC}(f_x, f_y)$ is a product of four components as shown below-

$$B_{DGC}(f_x, f_y) = K_{(x,y)} \times e^{-\frac{(I_x - I_y)^2}{2\sigma^2}} \times \frac{1}{d(x,y)} \times \frac{1}{\delta(x,y)_{DGC}} \quad (7.16)$$

where $d(x, y)$ represents the Euclidean distance between two voxels x and y having intensity values I_x and I_y respectively. The term $K_{(x,y)}$ is based on the probabilities of x and y to have the labeling f_x and f_y and is mathematically represented as:

$$K_{(x,y)} = 1 - |Pr(f_x)_{UN} - Pr(f_y)_{UN}| \quad (7.17)$$

where $f_x = f_y$. The factor σ is the standard deviation of voxel intensities of the image [294]. The term $\delta(x, y)_{DGC}$ denotes the sum of differences between probabilities of neighbouring voxels x and y to belong to f_x and f_y where $f_x \neq f_y$. This can be expressed as:

$$\begin{aligned} \delta(x, y)_{DGC} = & |Pr(f_x = \alpha)_{UN} - Pr(f_y = \alpha)_{UN}| \\ & + |Pr(f_x = \beta)_{UN} - Pr(f_y = \beta)_{UN}| \end{aligned} \quad (7.18)$$

where $\alpha, \beta \in \mathcal{L}$. The data term $R_{DGC}(f_x)$ is dependent on the probability map of UNet as shown below-

$$R_{DGC}(f_x) = -\ln Pr(f_x = \alpha)_{UN} \quad (7.19)$$

7.5.3 Alpha Beta Swap

We now discuss how and why we modify the optimization function of α - β swap moves. The choice of which type of move to select depends on whether the smoothness term of the energy function is a metric or a semi-metric [118]. If the smoothness

term is metric, α -expansion can be used, otherwise α - β swap move needs to be used. For a function $V(\alpha, \beta)$ to be metric, it has to satisfy the following constraints:

1. $V(\alpha, \beta) \Leftrightarrow \alpha = \beta$
2. $V(\alpha, \beta) = V(\beta, \alpha) \geq 0$
3. $V(\alpha, \beta) \leq V(\alpha, \gamma) + V(\gamma, \alpha)$

for any labels $\alpha, \beta, \gamma \in \mathcal{L}$ [118]. If $V(\alpha, \beta)$ satisfies only the constraints (1) and (2) but not (3), then it is called a semi-metric. We have chosen α - β swap moves to optimize our energy function, as our smoothness term ($B_{DGC}(f_x, f_y)$) is a semi-metric. We explicitly show in the appendix 7.9 that the smoothness term is indeed a semi-metric.

7.5.4 Deep learned Alpha Beta Swap

If a move from a partition V_l to a new partition V_l' has labels α, β , then $V_l' = V_l$ for all labels $l \neq \alpha, \beta$. This is known as a α - β swap [118]. The difference between V_l and V_l' is that some voxels that were labeled as α in V_l are now labeled as β , and the other way around. The main idea is to use graph cuts to separate all α voxels from β voxels one by one. Each time through the algorithm, the $\alpha - \beta$ mix will be different. The program will keep going through all the possible combinations until it converges with the minimum energy. The algorithm is guaranteed to converge in $O(V)$ time, but when there are a lot of vertices, the whole segmentation process takes a long time.

As reported in [118], segmenting a 384×288 image with $\alpha - \beta$ swap takes 35 seconds. In our case, the image size is $182 \times 218 \times 182$ which is far greater than the images used in [118]. Hence, there is a dire need to optimise the move algorithm to speed up the overall segmentation.

For this, we turn to Pseudo-Boolean optimization techniques used in [302]. As mentioned in [302], we encode the moves of the $\alpha - \beta$ swap algorithm as a vector of binary variables $t = t_i, \forall_i \in V$. $t_i = 0$ means the label of voxel i changed to α and $t_i = 1$ means the label changed to β . The transformation function $T(f^c, t)$ of a move

algorithm takes the current labelling f^c and a move t and returns a new labelling f^n that has been induced by the move. The transformation function $T_{\alpha\beta}()$ for an $\alpha - \beta$ swap transforms f^c as

$$f_i^n = T_{\alpha\beta}(f_i^c, t_i) = \begin{cases} f_i^c, & \text{if } f_i^c \neq \alpha \text{ and } f_i^c \neq \beta, \\ \alpha, & \text{if } f_i^c = \alpha \text{ or } \beta \text{ and } t_i = 0, \\ \beta, & \text{if } f_i^c = \alpha \text{ or } \beta \text{ and } t_i = 1. \end{cases} \quad (7.20)$$

If the current labelling f_i^c is neither α nor β , we don't change it. The energy of the move t is the energy of labelling f^n that the move t induces, *i.e.*, $E_m(t) = E(T(f^c mt))$. Further details about the pseudo boolean energy of the swap move can be found in Sec. 3.3 of [303].

We modify Eq. 7.20 by adding the label probability information derived from UNet as follows-

$$f_i^n = T(f_i^c, t_i) = \begin{cases} f_i^c, & \text{if } f_i^c \neq \alpha \text{ and } f_i^c \neq \beta, \\ \alpha, & \begin{cases} \text{if } f_i^c = \alpha \text{ or } \beta, t_i = 0, [Pr(f_x = \alpha) - Pr(f_x = \beta) > \tau] \\ \text{, or } [Pr(f_x = \beta) - Pr(f_x = \alpha) < (1 - \tau)], \end{cases} \\ \beta, & \begin{cases} \text{if } f_i^c = \alpha \text{ or } \beta, t_i = 1, [Pr(f_x = \beta) - Pr(f_x = \alpha) > \tau] \\ \text{, or } [Pr(f_x = \alpha) - Pr(f_x = \beta) < (1 - \tau)], \end{cases} \end{cases} \quad (7.21)$$

We added two more constraints when deciding the new labelling to be α or β . We wanted the confidence of the UNet model to decide whether a label should be swapped or should be kept the same. We define confidence of prediction as the difference between the probabilities of a voxel to have label α and β , *i.e.*, $[Pr(f_x = \alpha) - Pr(f_x = \beta)]$. Generally, a model is said to predict a label (say α) with high confidence if the probability of the voxel to belong to α is much higher than that of the voxel to belong to another label (say β), *i.e.*, if $[Pr(f_x = \alpha) - Pr(f_x = \beta)] \geq \tau$

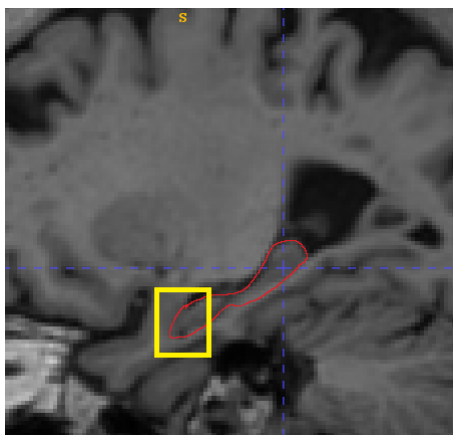


Figure 7.4: A sagittal slice view of a brain MRI showing the hippocampus bounded in red. The region marked in yellow shows that the contrast is less between hippocampus and its surrounding region, which poses a challenge in segmentation.

or $[Pr(f_x = \beta) - Pr(f_x = \alpha) \leq (1 - \tau)]$ where τ is some threshold. So, if for any voxel i , $[Pr(f_x = \alpha) - Pr(f_x = \beta)] \geq \tau$ and f_i^c , the current label of i is either α or β , then the label of voxel i will be changed to α . Similarly, the decision to change a label to β if the confidence of the UNet model for that voxel to have a label β than that of having label α is greater than τ .

7.5.5 Deep Learned Shape Information

In this section, we first discuss what is the significance of addition of a shape term in HC segmentation, then we briefly mention the importance of adaptive shape term and how the incorporation of UNet's probability map helps in creating an adaptive shape term suitable for 3D HC segmentation.

7.5.6 Need of an Adaptive Shape term

In cases where images are affected by substantial noise and intensity variations, the necessity for a shape prior can vary across different pixels. Consequently, assigning a uniform weight to the shape prior term for all pixels may not be suitable. In our case, 3D MRI images do suffer from noise and intensity variations and in many places the HC and non HC region of the brain has very low contrast as shown in Fig. 7.4. Segmentation tasks use the adaptive shape term to selectively impose shape

constraints based on pixel labeling difficulty to give flexibility and local adaptation. This adaptability allows the system to modify the strength of the shape prior based on local image properties, applying shape restrictions where they are most useful. The adaptive shape term dynamically adjusts shape priors based on image intensity, resulting in more accurate and context-aware segmentation results [292].

(A) Improved Adaptive Shape Term with UNet

We improve Wang *et al.*'s [292] adaptive shape prior formulation using learned information from 3D UNet. Following their approach, we add to the smoothness term $B_{DGC}(x, y)$, a shape term of the form $S_{DGC}(x, y)$ with η as the shape weight. Note that the authors in [292] defined $\eta = e^{-(Pr(x)-Pr(y))^2}$. $Pr(k)$ as the likelihood of a pixel k belonging to the foreground. They determined this likelihood by using an unsupervised technique like applying a Gaussian filter.

Unlike in [292], where the authors used 2D images and performed binary segmentation, we deal with 3D images and multi-class segmentation in this work. So we redefine η as:

$$\eta = e^{-(Pr(f_x)-Pr(f_y))^2} \quad (7.22)$$

where, $Pr(f_k)$ denotes the likelihood of voxel k to have labelling f_k . Further, $Pr(f_k)$ is obtained from the probability map of UNet as mentioned in Sec. 7.5.1. This ensures that we have better probability values than that obtained from using unsupervised techniques, as in [292].

The shape term, S_{DGC} , can be formulated as the unsigned distance function (as used in [304]) of the segmentation obtained after thresholding probability map \mathcal{P} . It is introduced to enforce structural consistency in the segmentation process by ensuring that the boundaries of the segmented object align with a shape prior. Let the segmentation obtained by thresholding \mathcal{P} is \mathcal{G} with a threshold value of κ . Then,

$$S_{DGC} = \bar{\phi}_{\mathcal{G}} \left(\frac{x+y}{2} \right) \quad (7.23)$$

where, $\bar{\phi}_{\mathcal{G}} : \mathfrak{R}^3 \rightarrow \mathfrak{R}$ is the distance function on \mathcal{G} and is such that $\bar{c} = \{x \in \mathfrak{R}^3 : \bar{\phi}(x) = 0\}$; \bar{c} being the set of points that form the boundary of the shape. The energy will be low if $\bar{\phi}_{\mathcal{G}}\left(\frac{x+y}{2}\right) \approx 0$ for all neighboring voxels x and y and $f_x \neq f_y$. If a voxel x lies near the shape template, then it will satisfy $\bar{\phi}(x) \approx 0$. Since, $\left(\frac{x+y}{2}\right)$ is roughly a point on the boundary of the segmented object, the condition for S_{DGC} to be small is the same as the condition that the boundary of the segmented object lies near the shape template.

7.5.7 Shape driven Multi-class Deep Graph Cut

We started with Eq. 7.14 which is the deep graph cut for energy function for binary segmentation. Then we modified it to adapt to multi-class segmentation in Eq. 7.24. We then modified the α - β swap moves using information from UNet as described in Sec. 7.5.4. We compute the data term $R_{DGC}(f_x)$, smoothness term $B'_{DGC}(f_x, f_y)$ and finally, after incorporating the two terms described in the previous section and shown in Eq. 7.22 and Eq. 7.23 in the energy function of multi-class deep graph cut (Eq. 7.15), we get the final energy function for Shape induced multi-class Deep Graph Cut (SMDGC) method as shown below-

$$\begin{aligned} \zeta_{MSDGC} = & \sum_{x \in X, y \in Ne(x), f_x \neq f_y} B_{DGC}(f_x, f_y) + \eta S_{DGC} + \\ & \lambda_{DGC}(f_x) \sum_{x \in X} R_{DGC}(f_x) \end{aligned} \quad (7.24)$$

The algorithm for our overall workflow is shown below in Algorithm 7.1 followed by a discussion on its time-complexity-

7.5.8 Time-complexity Analysis

The alpha-beta swap algorithm, used for multi-label graph cuts, involves iteratively optimizing the graph by swapping labels between pairs (α, β) to minimize the energy function. The time-complexity of this approach is influenced by several factors, including the number of labels, the number of pixels (or nodes), and the underlying max-flow algorithm used. Here's a breakdown of the time-complexity:

Algorithm 7.1: SMDGC

Input: 3D UNet model M trained on training set of the data, Graph G represented as a 3D grid of voxels

Output: G_{out} with desired segmentation and same dimensions as G

- 1 Compute data term $R_{DGC}(f_x)$ for each voxel in $x \in G$ as shown in Equation 7.19
- 2 Compute smoothness term $B_{DGC}(f_x, f_y)$ for each voxel x and its neighbor $y \in N_e(x)$ in G as shown in Equation 7.16
- 3 Compute η and S_{DGC} as shown in Equations 7.22 and 7.23 respectively
- 4 Compute the modified transformation function for optimizing the number of α - β swap moves using Equation 7.21.
- 5 Compute the final energy function using Equation 7.24, perform Graph cut and store the result in G_{out} .
- 6 **return** G_{out}

1. Max-flow Computation: The time-complexity of each max-flow computation depends on the specific max-flow algorithm used. We have used Edmond Karp’s implementation [305] of the Ford-Fulkerson Method [88] in our case. It has a time-complexity of $O(V \cdot E^2)$, where the symbols have usual meanings mentioned in Sec. 7.5.1.

2. Number of Labels: The alpha-beta swap involves considering all pairs of labels, so the number of iterations is proportional to $\binom{|\mathcal{L}|}{2} = \frac{|\mathcal{L}|(|\mathcal{L}|-1)}{2}$, which is $O(|\mathcal{L}|^2)$, where \mathcal{L} is the set of all labels.

Combining these factors, the overall time-complexity can be expressed as:

$$O(\mathcal{L}^2 \cdot V \cdot E^2) \tag{7.25}$$

7.6 Data preparation

In this section, we first discuss the datasets used and the data preparation steps taken for whole HC and HC subfield segmentation.

7.6.1 Data Preparation

Two publicly available datasets, namely, HarP [306] and Kulaga-Yoskovitz [307] were used for our experiments. Both of them contain T1-weighted MRI volumes with

ground truth. HarP is a protocol for manual HC segmentation that uses patient data from Alzheimer’s Disease Neuroimaging Initiative (ADNI) [308]. The complete release of HarP includes 135 T1-weighted MRI volumes which we divided into 80% training, 10% validation, and 10% testing. Kulaga-Yoskovitz (KY) dataset comprises 25 participants aged between 21 and 53 years, with a mean age of 31.2 ± 7.5 years and a male-to-female ratio of 12 : 13. Each participant data has manually segmented labels. The Montreal Neurological Institute (MNI) and Hospital’s Ethics Committee authorized the study. The data were acquired using a 3T Siemens Tim Trio MRI scanner equipped with a 32-channel head coil. Submillimeter T1 and T2 images were obtained for all participants. The 3D MPRAGE T1 image had a spatial resolution of $0.6 \times 0.6 \times 0.6 \text{mm}^3$ (isotropic voxel size). The matrix size was 336×384 , with a field of view (FOV) of $201 \text{mm} \times 229 \text{mm}$ and 240 axial slices at a slice thickness of 0.6mm . The manual segmentation protocol for this dataset categorized the hippocampus into three labels: subiculum (SUB), a combination of CA1, CA2, and CA3 (CA1-3), and a combination of CA4 and DG (CA4/DG). The KY dataset was evaluated for our investigation using a 5-fold cross validation. Different datasets have different voxel spacings. To standardize the voxel spacing in the dataset, we registered the images and transformed the labels corresponding to MNI template ($1 \times 1 \times 1 \text{mm}^3$ T1-w, dimensions: $192 \times 224 \times 192$, skull stripped) using FSL [272]. Then we skull stripped all the volumes and performed n4 bias field correction. Finally, we normalize all input images to have zero mean and unit standard deviation (std) based on non-zero voxels only. We have merged the ground truths of left and right HC into a single volume and trained our models to predict both the HC at once. In some ground truth volumes, we found holes inside the hippocampus segmentation that arised due to registration. We closed them using morphological closing operators. For HC subfield segmentation, we used only KY dataset preprocessed using the steps described in [282] that included cropping along the HC area and data augmentation by left - right flipping. We got 50 samples having 100 axial slices with the length and breadth same as mentioned earlier for T1 and T2 images. We could not use the HarP dataset as it did not have ground truth annotations for subfields.

7.7 Implementation Details

We implemented our networks in PyTorch [161]. We implemented our 3D UNet network in PyTorch [161]. The training process was executed on an HP-Z640 workstation having Intel Xeon processor with 14 Cores, a Random Access Memory (RAM) with capacity of 128GB along with a dedicated graphics processor unit (GPU) of 24GB with model name NVIDIA Titan RTX. Dice Similarity Score (DSC) is used as the performance metric [126] as this is the common metric used by all other competing methods. The higher the value of DSC, the better is the segmentation.

For whole HC segmentation: The cell size n (in Sec. 7.4.4) is set as 4. The parameter λ (Eq. 7.10) is gradually increased from 0.1 to 0.5 in each epoch of training. This is done to prevent the pooled gradient loss from incurring a huge penalty at the beginning. We make the model learn the coarse HC region in the beginning and then the fine HC shape at a later stage. The parameter β (in Sec. 7.4.4) is set to $\pm 12^\circ$. For this work, we divide a circle (see Fig. 7.3) into 8 equal sectors and consider α as the central angle. For example, if α is 22.5° , then all angles within the range $[22.5^\circ - 12^\circ, 22.5^\circ + 12^\circ]$ are considered. For the Attention Unet model, we used Adam optimizer with a batch size of 8. We trained the model for 200 epochs with our custom loss function (in Eq. 7.10).

For HC subfield segmentation The network is trained for 100 epochs with initial learning rate of 0.0001, weight decay of 0.00001 and mini-batch size equal to 2 samples. We have used Adam Optimizer and dice loss in the process. Our energy function does not have any parameters that need to be set manually, as all the information is being provided by a trained UNet model. The only parameter κ , which is used for thresholding the UNet probabilities to create a segmentation (as described in Sec. 7.5.6) is set to, 0.5 which is the most common value as mentioned in [309].

Table 7.1: Comparison with state-of-the-art methods

Approach	DSC		Time (in secs)
	Harp	KY	
Level Set (2019) [274]	0.847	-	-
MAS (2021) [273]	0.885	-	8.88
Hippodeep (2018) [310]	0.85	0.82	2
SWANS (2022) [311]	0.878	-	20
Hippmapp3r (2019) [277]	0.87	0.836	14
E2DHipSeg (2021) [312]	0.88	0.84	15
Subfield (2022) [313]	0.934	0.902	-
Shape SDM (2020) [281]	0.843	-	-
Shape fitting (2020) [280]	0.856	0.92	150
Our Method	0.947	0.923	12

Table 7.2: Ablation study for the proposed loss function

Loss Function	DSC	
	Harp	KY
Dice Loss	0.85	0.835
Dice + Pooled Gradient Loss	0.947	0.923

7.8 Experimental Results

In this section, we discuss the ablation studies conducted for whole HC segmentation and HC subfield segmentation, followed by comparison with state-of-the-art methods for both the segmentation tasks.

7.8.1 Whole Hippocampus Segmentation

(A) Ablation Study

We have conducted an ablation study to demonstrate the effectiveness of the proposed loss function with that of the 3D Attention Unet having only Dice loss. The DSC values in Table 7.2 clearly indicate that our proposed loss function has achieved superior results on both the datasets over the Dice loss alone. Qualitative results for a sample from the HarP dataset, as shown in Fig. 7.5, corroborate the same.

Table 7.3: Ablation Study I: Comparison of segmentation performance of multi-class graph cut, multi-class UNet, and the proposed method. Mean Dice Score of each competing approach over all three classes are reported. Best values are shown in **bold**.

Method	Dice Score
Multi-class Graph Cut	0.64000 \pm 0.073
Multi-class UNet	0.82000 \pm 0.047
SMDGC (Ours)	0.91467 \pm 0.009

(B) Comparisons with Other Methods

We have compared our method with as many as nine state-of-the-art segmentation approaches from three different categories and the results are shown in Table 7.1. Among these nine methods, two did not use any DL (marked in blue), five used DL with regular loss functions like Dice or cross-entropy (marked in green) and two applied DL with shape based loss functions (marked in red). Some methods like Hippodeep, Hippmapp3r, E2DHipSeg and Subfield have used multiple datasets. Hence, we re-implemented those works using HarP and KY datasets with their publicly available codes. However, we left some blank entries in the KY column as the corresponding works did not report Dice Scores for the KY dataset and had no publicly available code for re-implementation. Results in the Table 7.1 clearly establish that the proposed method achieves state-of-the-art results by outperforming all other competing methods and ranking third best in terms of the execution time.

7.8.2 Hippocampus Subfield Segmentation

(A) Ablation Studies

As mentioned earlier, we present three ablation studies for providing a better understanding of our solution. Table 7.3 shows the results of our first ablation study. Here, we demonstrate the improvement our model brings over baseline multi-class graph cut and multi-class UNet, applied in isolation. In Table 7.4, we first show the performance of DGC [83] for multi-class segmentation, by adding to it traditional α - β swap. It is then demonstrated how the execution time improves due to

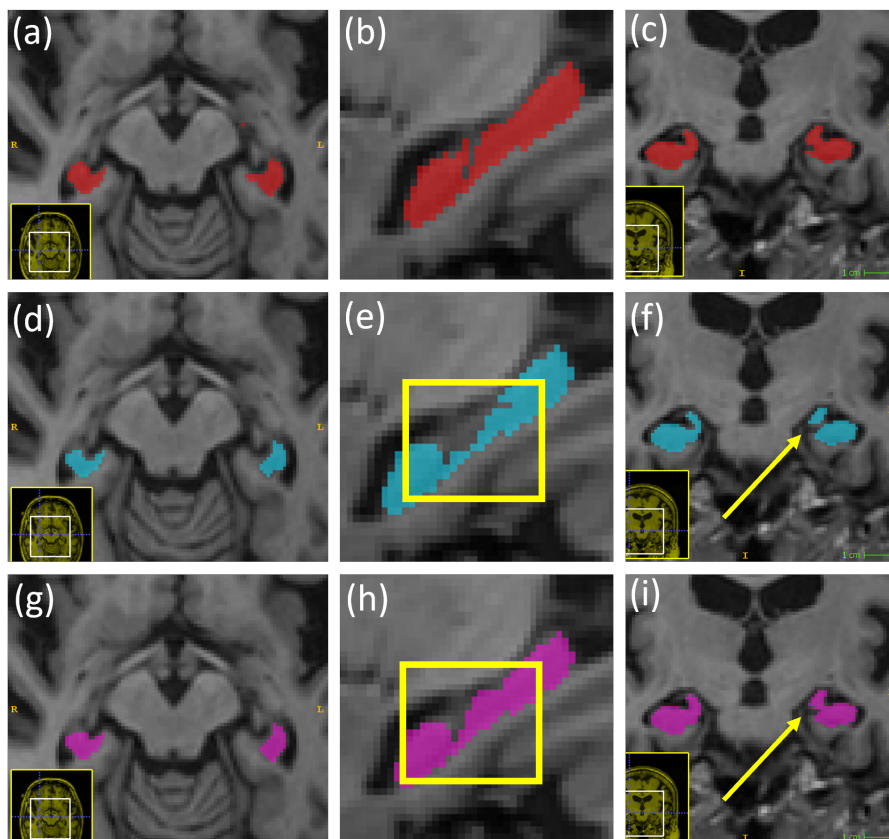


Figure 7.5: Visual comparisons of ground truth segmentation with a model having only dice loss and the same model having the proposed loss function: (a), (b) and (c) represent the axial, sagittal and coronal views of the ground truth HC segmentation, (d), (e) and (f) represent the same views of the predicted segmentation output of the model with dice loss only, (dice score: 0.848), (g), (h) and (i) represent the same views of the predicted segmentation of our proposed loss function (dice score: 0.927). The difference in prediction can be seen in (e) vs. (h) and in (f) vs. (i) and is highlighted with yellow squares and yellow arrows respectively.

Table 7.4: Ablation Study II: Impact of deep learned α - β swap on the segmentation performance. Mean Dice Score of each competing approach over all three classes are reported. Best values are shown in **bold**.

Method	Dice Score	Time (in secs)
DGC [83] with α - β swap	0.88230 \pm 0.032	15
DGC [83] with deep learned α - β swap	0.89860 \pm 0.008	8

Table 7.5: Ablation Study III: Impact of deep learned shape on segmentation performance. Mean Dice Score of each competing approach over all three classes are reported. Best values are shown in **bold**.

Method	Dice Score
Graph cut with adaptive shape term [292]	0.85297 \pm 0.026
SMDGC (Ours)	0.91467 \pm 0.009

optimization of α - β swap strategy using learned information from UNet. The computation time improves drastically improves by almost 50% when we use modified α - β swap. DGC with normal α - β swap takes on average 15 seconds to segment one 3D sample, whereas, the modified α - β swap achieves the same goal in 8 seconds. A slight improvement in the segmentation accuracy can also be noticed as more informed decision is taken to change a label during a move due to the confidence of UNet incorporated into the optimization strategy (Eq. 7.21).

We then analyze the impact of a shape term in a graph cut setup through Table 7.5. The first row shows the results, where an adaptive shape term is used but without deep learning. For that, we re-implement [292] and add α - β swap moves, as that work was originally developed for binary segmentation. We compare the performance of this method with ours, where we have employed deep learned shape information (Eq. 7.24). The values of the Dice Scores clearly illustrate the benefits of a deep learned shape information.

As we can see, our proposed method performs better than rest of the ablation entries and we can also see that each component has a major role to play in the overall segmentation. Also, the segmentation computation time for our proposed method is 8 seconds on average. Qualitative comparisons of different strategies used in the three ablation studies are shown in figure 7.6. We only include the second method of Table 7.4 as the improvement there is more in terms of execution time to achieve desired segmentation, rather than the segmentation accuracy per se. In Fig. 7.6, the visual improvements in segmentation performances clearly corroborate the quantitative results. We specifically highlight how multi-class UNet and graph cut with adaptive shape term suffers from over segmentation of CA1-3 and SUB, as shown in the yellow boxes of the sagittal slices. DGC with modified α - β swap

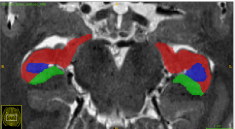
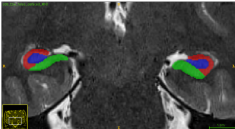
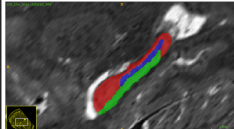
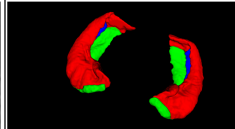
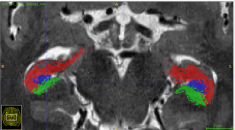
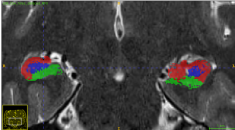
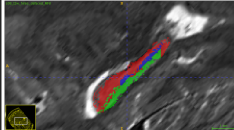
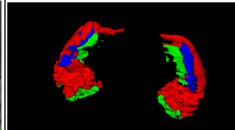
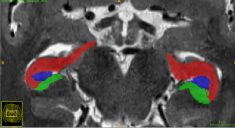
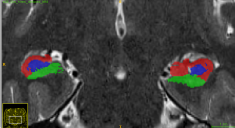
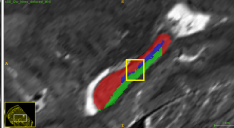
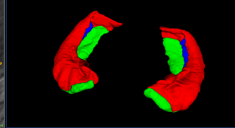
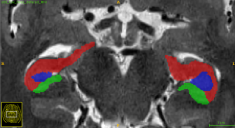
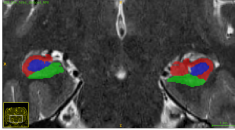
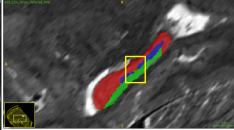
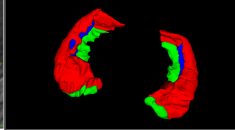
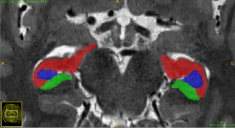
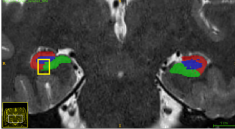
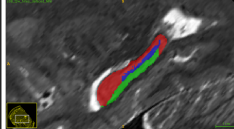
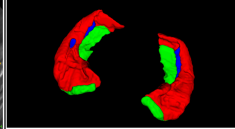
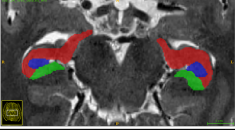
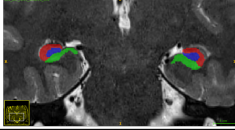
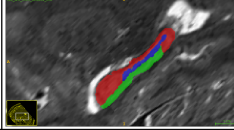
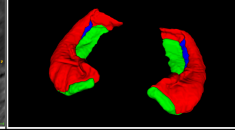
	Axial	Coronal	Sagittal	3D
Ground truth				
Graph cut multi class (DSC 0.573)				
UNet multi class (DSC 0.812)				
Graph cut with shape [30] (DSC 0.859)				
DGC with modified α - β swap (DSC 0.8854)				
SMDGC (Ours) (DSC 0.9086)				

Figure 7.6: Qualitative ablation of our method. GT represents the ground truth. Segmentation with color red represents CA1-3 class, blue represents the CA4/DG class and green represents the SUB class.

Table 7.6: Comparison with state-of-the-art methods. Mean Dice Score \pm standard deviation of Dice Score is reported for each class. Additionally, the overall mean Dice Score is reported for each method. Best values are shown in **bold**.

Method	CA1-3	CA4/DG	SUB	Mean
Syn SegNet (2023) [287]	0.865 \pm 0.005	0.821 \pm 0.014	0.821 \pm 0.013	0.835
CAST (2020) [286]	0.917 \pm 0.011	0.89 \pm 0.017	0.881 \pm 0.021	0.906
ResDUNet (2019) [283]	0.92 \pm 0.011	0.879 \pm 0.02	0.888 \pm 0.018	0.896
UNet CNN (2022) [282]	0.9245 \pm 0.01	0.8887 \pm 0.023	0.898 \pm 0.015	0.9
SMDGC (Ours)	0.933 \pm 0.007	0.9078 \pm 0.013	0.903 \pm 0.008	0.9146

also finds it difficult to decide among the CA1-3 and SUB, as shown in the yellow box of its coronal slice. In general, CA4/DG is relatively difficult to segment by all methods, as it is the smallest region among the three classes under consideration.

(B) Comparison with State-of-the-art Methods

We compare our proposed method with five state-of-the-art approaches (articles published within the last five years). These methods are [287], [286], [283], [282], and, [284]. We showed comparisons with only DL based approaches, as we did not come across any work on multi-class HC segmentation using primarily graph cuts. As can be clearly seen from the Table 7.6, our method has yielded the highest mean Dice Score, which is marginally better than [284]. We are marginally behind [284] in the SUB subfield segmentation, the most complex object to segment within HC. However, our model requires much less computational resource, as we used plain 3D UNet & graph cut while, other approaches used sophisticated DL models that takes more time & resources to train.

7.9 Appendix

Proof of Lemma 1: The product of a semi-metric and a metric function is semi-metric.

Proof. Let ρ_1 be a semi-metric function and ρ_2 be a metric function defined on some set X . Then, for any $x, y, z \in X$:

$$\begin{aligned}\rho_1(x, z) &\leq \rho_1(x, y) + \rho_1(y, z) \quad (\because \rho_1(x, z) \text{ is semi-metric}) \\ &= c_1 \cdot (\rho_1(x, y) + \rho_1(y, z)) \quad (\text{where } c_1 \leq 1)\end{aligned}$$

$$\begin{aligned}\rho_2(x, z) &> \rho_2(x, y) + \rho_2(y, z) \quad (\because \rho_2(x, z) \text{ is metric}) \\ &= c_2 \cdot (\rho_2(x, y) + \rho_2(y, z)) \quad (\text{where } c_2 > 1)\end{aligned}$$

Now, consider the product

$$\begin{aligned}\rho(x, z) &= \rho_1(x, z) \cdot \rho_2(x, z) \\ &= \left(c_1 (\rho_1(x, y) + \rho_1(y, z)) \right) \cdot \left(c_2 (\rho_2(x, y) + \rho_2(y, z)) \right) \\ &= c_1 c_2 \rho_1(x, y) \rho_2(x, y) + c_1 c_2 \rho_1(x, y) \rho_2(y, z) \\ &\quad + c_1 c_2 \rho_2(y, z) \rho_1(x, y) + c_1 c_2 \rho_2(y, z) \rho_1(y, z)\end{aligned}$$

Now, $c_1 c_2 > 1$ when $c_1 = 1$. Hence,

$$\begin{aligned}\rho(x, z) &> \rho_1(x, y) \rho_2(x, y) + \rho_1(y, z) \rho_2(y, z) + \rho_1(x, y) \rho_2(y, z) + \rho_2(y, z) \rho_1(x, y) \\ &> \rho(x, y) + \rho(y, z) + \omega_1 + \omega_2\end{aligned}$$

where $\omega_1 \geq 0$ and $\omega_2 \geq 0$.

Therefore, $\rho(x, z) > \rho(x, y) + \rho(y, z)$ when $c_1 = 1$ which means $\rho(x, z)$ does not obey triangle inequality for some particular cases. Thus, the product of a semi-metric and a metric function remains a semi-metric function.

Proof of Theorem: $B_{DGC}(f_x, f_y)$ is a semi-metric.

Proof. B_{DGC} is a product of four components as shown in Eq. 7.16. Among them, $K_{(x,y)}$ and $\delta(x,y)_{DGC}$ depends on probabilities. From Eq. 7.17, it is evident that $K_{(x,y)}$ lie between $[0, 1]$ whereas, from Eq. 7.18, $\delta(x,y)_{DGC}$ lie between $[0, 2]$. Both these functions satisfy points (1) and (2) of Sec. 7.5.3, *i.e.*, $K_{x,y} \Leftrightarrow x = y$, $K_{x,y} = K_{y,x} \geq 0$ and similarly for $\delta(x,y)_{DGC}$. But they do not satisfy the triangle inequality (point 3). If we consider three voxels x, y and z , then $K_{x,y}$, $K_{y,z}$ and $K_{x,z}$ can take any value between $[0, 1]$ and hence, there will be cases where $K_{(x,z)} > K_{(x,y)} + K_{(y,z)}$ for some x, y and z . A similar situation can also occur in the case of $\delta(x,y)_{DGC}$. Therefore, these functions are semi metric. For example, if we consider $K_{(x,y)} = 0.2$, $K_{(y,z)} = 0.3$ and $K_{(x,z)} = 0.7$, then $K_{(x,z)} > K_{(x,y)} + K_{(y,z)}$. Now, we consider the term $e^{-\left(\frac{(I_x - I_y)^2}{2\sigma^2}\right)}$, which is based on image intensities I_x and I_y . The intensity value lies between $[0, 255]$. We can similarly argue that $e^{-\left(\frac{(I_x - I_z)^2}{2\sigma^2}\right)} > e^{-\left(\frac{(I_x - I_y)^2}{2\sigma^2}\right)} + e^{-\left(\frac{(I_y - I_z)^2}{2\sigma^2}\right)}$ for some I_x, I_y and I_z . Thus, $K_{(x,y)}$, $e^{-\left(\frac{(I_x - I_y)^2}{2\sigma^2}\right)}$ and $\delta(x,y)_{DGC}$ are semi metric in nature, $\frac{1}{d(x,y)}$ is metric as $d(x,y)$ is the Euclidean distance. Hence, from Lemma 7.9 it follows that $B_{DGC}(x,y)$ is a semi-metric.

7.10 Summary

Segmentation of the hippocampus and its subfields is vital for diagnosing diseases like Alzheimer's and epilepsy, as treatment relies on analyzing subfield atrophy. Automating this process can significantly improve the treatment experience for both doctors and patients. We propose advanced methods for whole and subfield segmentation using a novel shape-driven loss function, a combination of multi-class graph cuts with shape information, and deep learning. Using HoG based loss function for whole hippocampus segmentation greatly improves the segmentation accuracy as it is able to learn the shape well. Using shape prior in energy function of multi-class graph cut helps to learn the shapes of the subfields as well. Our approach outperforms existing methods, as demonstrated by comparisons on a public dataset.

Chapter 8

Conclusion

In this thesis, we have addressed several problems in diagnosing neurological disorders using computer vision. We have provided a detailed description of proposed methods in the preceding chapters, with relevant qualitative and quantitative results to validate the effectiveness of our solutions. In this chapter, we conclude the work carried out in this thesis by highlighting the contributions, and then indicate possible future research directions.

8.1 Concluding Remarks

The thesis consists of a total of eight chapters. We have three chapters on Introduction (Chapter 1), Theoretical Foundations (Chapter 2), and Conclusion (this chapter). The rest five chapters, Chapters 3 to Chapter 7, are devoted to specific research problems in neuro-radiology.

We first focused on the brain tumors. Here, we primarily concentrated on the detection, delineation, and characterization of 3D brain tumors using MR imaging, which is crucial in guiding treatment strategies. We demonstrated how combining UNet and graph cut achieves better segmentation performance in 3D. We derived new expressions for the constituent terms in the graph cut energy function using probability maps from the UNet. Through comprehensive experimentation, we established that our proposed deep graph cut model yields competitive performance on the publicly available BRATS dataset. Then we presented a brain tumor clas-

sification strategy by combining deep features and Graph Convolutional Networks (GCN). We showed that using both radiology and histopathology data achieves state-of-the-art performance.

We next created a genetic marker prediction model that can predict the status of five important bio markers (IDH, 1p/19q codeletion status, ATRX, MGMT, and TERT), which are responsible for causing different types of brain tumors, from Whole Slide Imaging (WSI) data. This simultaneous prediction ensures better prognosis and treatment planning for patients. We harnessed different prevalent traits (individual, pairwise, and group-wise behavior) in these biomarkers, as reported in the scientific literature. We integrated conditional probability-based loss capturing pairwise information and spectral graph-based loss modeling group behavior with a multi-label weighted cross-entropy loss for individual traits (for the five genetic markers) in the form of a composite loss function. Additionally, we created a benchmark dataset by integrating WSI and biomarker data, serving as a helpful testbed for future research.

After that, we shifted our focus to classification of Alzheimer’s Disease (AD), which is crucial for dementia diagnosis, especially in the aging population. We introduced an automated solution for four-class classification of AD using efficient processing of 3D Diffusion Tensor Imaging (DTI) data. Initially, we trained separate deep learning (VoxCNN) and machine learning (Random Forest) models on different pieces of information in DTI scan volumes. Then, we combined the individual classification results using a modulated rank averaging decision fusion strategy. Comprehensive experimentation on the ADNI database demonstrated the effectiveness of our formulation, including comparisons and ablation studies.

Finally, we addressed the challenging problem of automated segmentation of the hippocampus (HC) and its subfields, which significantly enhances the efficiency of neuro-radiologists and aids treatment planning for neuro-surgeons, playing a vital role in diagnosing and managing diseases like Alzheimer’s and Epilepsy. We introduced a Histogram of Oriented Gradients (HOG) based loss function in an Attention UNet for 3D segmentation of the HC, addressing the complex and irregular structure

of the brain. To improve segmentation accuracy, we introduced a pooling function to resolve the derivability issue of histograms in HOG. Additionally, we proposed a state-of-the-art method for subfield segmentation, combining multi-class graph cuts with shape information and deep learning, showcasing how deep learning enhances shape knowledge and optimizes the $\alpha - \beta$ swap move. Our approaches outperformed several state-of-the-art methods on publicly available datasets.

8.2 Graphical User Interface

We also developed a Graphical User Interface (GUI) application called "Brain Analysis and Visualization App" to run some of our developed works, namely, 3D Brain tumor segmentation and 3D Hippocampus Segmentation and which are described in Chapter 3 and 7 respectively. Fig. 8.1 shows some screenshots of the application, where a brain scan of a patient with a tumor is uploaded, and the tumor is segmented.

8.3 Future Directions

Although we have addressed and proposed innovative solutions to challenging problems in the detection and diagnosis of neurological disorders, several advancements in these directions can still be made.

Future research in brain tumor segmentation can involve incorporating more recent datasets with various modalities like Computed Tomography and advanced versions of MRI. More sophisticated performance metrics like Hausdorff distance can be introduced for better analysis of segmentation performance. The proposed approach can also be extended for multi-label tumor segmentation. With regard to brain tumor classification, other modalities and tumor grades can be included to perform a more comprehensive classification.

Future research on genetic biomarker detection will focus on incorporating explainability [314] into our proposed methods to make the model more trustworthy and transparent for medical practitioners. The present model can be extended to

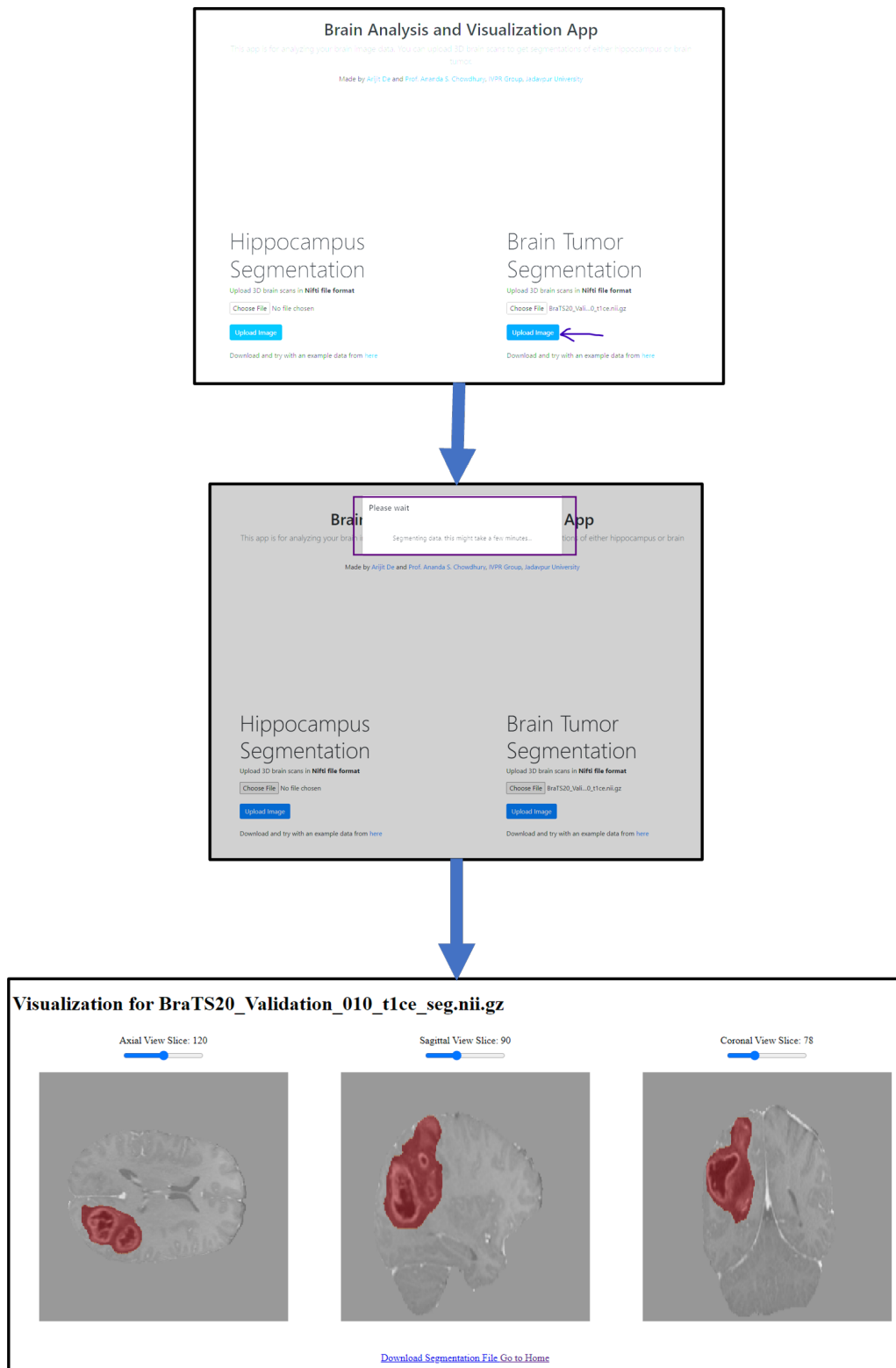


Figure 8.1: Screenshots of the GUI showing the steps of uploading a brain scan, segmentation and then visualizing the segmented tumor

predict other important bio markers of the brain.

The Alzheimer's Disease classification model can be extended to solve a six-class AD classification problem by incorporating Mild Cognitive Impairment and Significant Memory Concern stages [315]. Another direction of future research will be to predict the next stage of AD progression with significant accuracy.

For Hippocampus segmentation, incorporation of shape-based loss functions to better detect structures of varying shapes and sizes can be introduced. Additionally, we will include more datasets with additional subfields to achieve more fine-grained segmentation of the hippocampus, ultimately improving the diagnostic and treatment processes for neurodegenerative conditions.

Bibliography

- [1] J. Campellone, M. Fraser, and R. K. Turley, “Alzheimer disease,” <https://www.cedars-sinai.org/health-library/diseases-and-conditions/a/alzheimers-disease.html>, [Accessed 19-03-2025].
- [2] N. Challenged, “Know your brain: Hippocampus,” <https://neuroscientificallychallenged.com/posts/know-your-brain-hippocampus>, [Accessed 19-03-2025].
- [3] D. Daimary, M. B. Bora, K. Amitab, and D. Kandar, “Brain tumor segmentation from mri images using hybrid convolutional neural networks,” *Procedia Computer Science*, vol. 167, pp. 2419–2428, 2020.
- [4] B. Zhao *et al.*, “Cnn-based fully automatic glioma classification with multi-modal medical images,” in *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*. Cham: Springer International Publishing, 2021, pp. 497–507.
- [5] R. Gauriau, “Shape-based approaches for fast multi-organ localization and segmentation in 3d medical images,” Ph.D. dissertation, Telecom Paris, 06 2015.
- [6] T. Collins, “Graph cut matching in computer vision,” *University of Edinburgh*, 2004.
- [7] E. Akgün and M. Demir, “Modeling course achievements of elementary education teacher candidates with artificial neural networks,” *International Journal of Assessment Tools in Education*, vol. 5, no. 3, pp. 491–509, 2018.

- [8] X. Yan, T. Ai, M. Yang, and H. Yin, “A graph convolutional neural network for classification of building patterns using spatial vector data,” *ISPRS journal of photogrammetry and remote sensing*, vol. 150, pp. 259–273, 2019.
- [9] K. Clark *et al.*, “The cancer imaging archive (tcia): maintaining and operating a public information repository,” *J. Digit Imaging*, vol. 26, pp. 1045–1057, 2013.
- [10] A. Arcella, F. Limanaqi, R. Ferese, F. Biagioni, M. A. Oliva, M. Storto, M. Fanelli, S. Gambardella, and F. Fornai, “Dissecting molecular features of gliomas: Genetic loci and validated biomarkers,” *Int. J. Mol. Sci.*, vol. 21, no. 2, p. 685, 2020.
- [11] Q. Ostrom, M. Price, C. Neff, G. Cioffi, K. Waite, C. Kruchko, and J. Barnholtz-Sloan, “Cbtrus statistical report: Primary brain and other central nervous system tumors diagnosed in the united states in 2016-2020,” *Neuro Oncol*, vol. 25, no. 12 Suppl 2, pp. iv1–iv99, 2023.
- [12] K. Yang, Z. Wu, H. Zhang, N. Zhang, W. Wu, Z. Wang, Z. Dai, X. Zhang, L. Zhang, Y. Peng *et al.*, “Glioma targeted therapy: insight into future of molecular approaches,” *Molecular Cancer*, vol. 21, no. 1, p. 39, 2022.
- [13] S. Grochans, A. M. Cybulska, D. Simińska, J. Korbecki, K. Kojder, D. Chlubek, and I. Baranowska-Bosiacka, “Epidemiology of glioblastoma multiforme—literature review,” *Cancers*, vol. 14, no. 10, p. 2412, 2022.
- [14] S. Liu *et al.*, “Isocitrate dehydrogenase (idh) status prediction in histopathology images of gliomas using deep learning,” *Sci. Rep.*, vol. 10, no. 1, p. 7733, 2020.
- [15] M. Szyłberg, P. Sokal, P. Śledzińska, M. Bebyn, S. Krajewski, Ł. Szyłberg, A. Szyłberg, T. Szyłberg, K. Krystkiewicz, M. Birski *et al.*, “Mgmt promoter methylation as a prognostic factor in primary glioblastoma: a single-institution observational study,” *Biomedicines*, vol. 10, no. 8, p. 2030, 2022.

- [16] P. Śledzińska, M. G. Bebyn, J. Furtak, J. Kowalewski, and M. A. Lewandowska, “Prognostic and predictive biomarkers in gliomas,” *Int. J. Mol. Sci.*, vol. 22, no. 19, p. 10373, 2021.
- [17] M. S. Tomar, A. Shrivastava *et al.*, “Tert promoter mutations correlate with idhs, mgmt and egfr in glioblastoma multiforme,” *Neurology India*, vol. 69, no. 1, p. 135, 2021.
- [18] D. Lin, M. Wang, Y. Chen, J. Gong, L. Chen, X. Shi, F. Lan, Z. Chen, T. Xiong, H. Sun *et al.*, “Trends in intracranial glioma incidence and mortality in the united states, 1975-2018,” *Frontiers in oncology*, vol. 11, p. 748061, 2021.
- [19] M. Tzioras, R. I. McGeachan, C. S. Durrant, and T. L. Spires-Jones, “Synaptic degeneration in alzheimer disease,” *Nature Reviews Neurology*, vol. 19, no. 1, pp. 19–38, 2023.
- [20] A. Nandi, N. Counts, S. Chen, B. Seligman, D. Tortorice, D. Vigo, and D. E. Bloom, “Global and regional projections of the economic burden of alzheimer’s disease and related dementias from 2019 to 2050: A value of statistical life approach,” *EClinicalMedicine*, vol. 51, 2022.
- [21] E. Nichols and T. Vos, “Estimating the global mortality from alzheimer’s disease and other dementias: A new method and results from the global burden of disease study,” *Alzheimers Dement.*, vol. 16, p. e042236, 2019.
- [22] C. D. Billones, O. J. L. D. Demetria, D. E. D. Hostallero, and P. C. Naval, “Demnet: A convolutional neural network for the detection of alzheimer’s disease and mild cognitive impairment,” in *2016 IEEE Region 10 Conference (TENCON)*, 2016, pp. 3724–3727.
- [23] K. S. Anand and V. Dhikav, “Hippocampus in health and disease: An overview,” *Annals of Indian Academy of Neurology*, vol. 15, no. 4, pp. 239–246, 2012.

- [24] C. R. Jack, D. S. Knopman, W. J. Jagust, L. M. Shaw, P. S. Aisen, M. W. Weiner, R. C. Petersen, and J. Q. Trojanowski, "Hypothetical model of dynamic biomarkers of the alzheimer's pathological cascade," *The Lancet Neurology*, vol. 9, no. 1, pp. 119–128, 2010.
- [25] Y. Babu, T. Sheleme, B. Sefera, F. Bekele, and W. Olike, "Magnitude of uncontrolled seizures and associated factors among people with epilepsy. data from a major teaching hospital in southwest ethiopia," *Epilepsy & Behavior*, vol. 140, p. 109089, 2023.
- [26] R. D. Walter, "Clinical aspects of temporal lobe epilepsy," *California Medicine*, vol. 110, no. 4, p. 325, 1969.
- [27] P. J. Basser, J. Mattiello, and D. LeBihan, "MR diffusion tensor spectroscopy and imaging," *Biophysical Journal*, vol. 66, no. 1, pp. 259–267, 1994.
- [28] B. Jeevanandham, T. Kalyanpur, P. Gupta, and M. Cherian, "Comparison of post-contrast 3d-t 1-mprage, 3d-t 1-space and 3d-t 2-flair mr images in evaluation of meningeal abnormalities at 3-t mri," *The British Journal of Radiology*, vol. 90, no. 1074, p. 20160834, 2017.
- [29] C. Muccio, A. Tarantino, G. Esposito, and A. Cerase, "Differential diagnosis by unenhanced flair t2-weighted magnetic resonance images between solitary high-grade gliomas and cerebral metastases appearing as contrast-enhancing cortico-subcortical lesions," *Journal of Neuro-Oncology*, vol. 103, pp. 713–717, 2011.
- [30] V. Mathews, K. Caldemeyer, M. Lowe, S. Greenspan, D. Weber, and J. Ulmer, "Brain: gadolinium-enhanced fast fluid-attenuated inversion-recovery mr imaging," *Radiology*, vol. 211, no. 1, pp. 257–263, 1999.
- [31] J. Martina, C. Simmons, and D. Jukic, "High-definition hematoxylin and eosin staining in a transition to digital pathology," *Journal of Pathology Informatics*, vol. 2, 2011.

- [32] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *Proc. CVPR*, 2017, pp. 4700–4708.
- [33] Ç. et al., “3d u-net: learning dense volumetric segmentation from sparse annotation,” in *MICCAI*. Springer, 2016, pp. 424–432.
- [34] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” in *ICLR Conference Track Proceedings*. OpenReview.net, 2017.
- [35] V. S. V. Anjali Wadhwa, Anuj Bhardwaj, “A review on brain tumor segmentation of mri images,” *Magnetic Resonance Imaging*, vol. 61, pp. 247–259, 2019.
- [36] R. E. W. Rafael C. Gonzalez, *Digital Image Processing*, 3rd ed. Prentice Hall, 2007.
- [37] T. Jiang *et al.*, “Region growing segmentation algorithm for mri image based on the neighborhood attraction,” *Computational and mathematical methods in medicine*, vol. 2013, pp. 1–10, 2013.
- [38] N. Gordillo, E. Montseny, and P. Sobrevilla, “State of the art survey on mri brain tumor segmentation,” *Magnetic resonance imaging*, vol. 31, pp. 1426–1438, 2013.
- [39] L. Deng and Y. Dong, “Deep learning: Methods and applications,” *Foundations and Trends in Signal Processing*, vol. 7, pp. 197–387, 2014.
- [40] H. M. *et al.*, “Brain tumor segmentation with deep neural networks,” *Med Image Anal*, vol. 35, pp. 18–31, 2017.
- [41] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.
- [42] e. a. Ahmed, M. N., “A modified fuzzy c-means algorithm for bias field estimation and segmentation of mri data,” *IEEE transactions on medical imaging*, vol. 21, pp. 193–199, 2002.

- [43] N. Parveen and I. Sathik, "A hybrid approach for segmentation of brain mri using fcm with svm," *International Journal of Computer Applications*, vol. 84, pp. 21–26, 2013.
- [44] J. Selvakumar, A. Lakshmi, and T. Arivoli, "Brain tumor segmentation and its area calculation in brain mr images using k-means clustering and fuzzy c-means clustering," in *Proceedings of the International Conference on Advances in Engineering, Science and Management*, 2012, pp. 186–190.
- [45] C. V. *et al.*, "Geometric deformable models," *Reference*, vol. 61, pp. 247–259, 2019.
- [46] R. V. Shree and T. N. Kumar, "Identification and classification of brain tumor mri images with feature extraction using dwt and probabilistic neural network," *Brain Informatics*, vol. 5, pp. 23–30, 2018.
- [47] U. Javed, M. Riaz, A. Ghafoor, and T. Cheema, "Mri brain classification using texture features, fuzzy weighting and support vector machine," *Prog. Electromagn. Res. B*, vol. 53, pp. 73–88, 2013.
- [48] H. Selvaraj, S. Selvi, D. Selvathi, and L. Gewali, "Brain mri slices classification using least squares support vector machine," *Int. J. Intell. Comput. Med. Sci. Image Process.*, vol. 1, pp. 21–33, 2007.
- [49] P. John, "Brain tumor classification using wavelet and texture based neural network," *Int. J. Sci. Eng. Res.*, vol. 3, pp. 1–7, 2012.
- [50] H. Greenspan, B. Van Ginneken, and R. Summers, "Guest editorial deep learning in medical imaging: overview and future promise of an exciting new technique," *IEEE Trans. Med. Imaging*, vol. 35, pp. 1153–1159, 2016.
- [51] G. Litjens, T. Kooi, B. Bejnordi, A. Setio, F. Ciompi, M. Ghafoorian, J. van der Laak, B. Van Ginneken, and C. Sánchez, "A survey on deep learning in medical image analysis," *Med. Image Anal.*, vol. 42, pp. 60–88, 2017.

- [52] S. Ren, K. He, R. Girshick, X. Zhang, and J. Sun, “Object detection networks on convolutional feature maps,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, pp. 1476–1481, 2017.
- [53] Y. Bengio, “Learning deep architectures for ai,” *Found. Trends Mach. Learn.*, vol. 2, pp. 1–127, 2009.
- [54] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, pp. 436–444, 2015.
- [55] M. Gao, U. Bagci, L. Lu, A. Wu, M. Buty, H.-C. Shin, H. Roth, G. Papadakis, A. Depeursinge, and R. Summers, “Holistic classification of ct attenuation patterns for interstitial lung diseases via deep convolutional neural networks,” in *Comput. Methods Biomech. Biomed. Eng. Imaging Vis.*, 2016, pp. 1–6.
- [56] H.-C. Shin, L. Lu, L. Kim, A. Seff, J. Yao, and R. Summers, “Interleaved text/image deep mining on a very large-scale radiology database,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1090–1099.
- [57] J. Margeta, A. Criminisi, R. Cabrera Lozoya, D. Lee, and N. Ayache, “Fine-tuned convolutional neural nets for cardiac mri acquisition plane recognition,” in *Comput. Methods Biomech. Biomed. Eng. Imaging Vis.*, vol. 5, 2017, pp. 339–349.
- [58] e. a. Kloten, “Promoter methylation of tumor suppressor genes in breast cancer,” *Breast Cancer Research*, vol. 15, p. R4, 2013.
- [59] e. a. Rehman, “Machine learning algorithms validate mirnas as breast cancer biomarkers,” *BMC Cancer*, vol. 19, p. 1234, 2019.
- [60] e. a. Ma, “Identification of gene prognosis markers for lung adenocarcinoma,” *Nature Communications*, vol. 11, p. 612, 2020.
- [61] e. a. Xing, “Identification of prognostic small nucleolar rnas in cancer,” *Scientific Reports*, vol. 10, p. 12345, 2020.

- [62] e. a. Wong, “Deep learning identifies prognostic genes for glioblastoma,” *Cancer Research*, vol. 80, pp. 1234–1245, 2020.
- [63] e. a. Nam, “Gene ranker method for identifying predictive biomarkers,” *Bioinformatics*, vol. 36, pp. 3456–3467, 2020.
- [64] e. a. Zhao, “Graph convolutional network for prioritizing genes,” *Bioinformatics*, vol. 36, pp. 123–134, 2020.
- [65] e. a. Zhang, “Network-based deep learning approach for gene prioritization,” *PLoS Computational Biology*, vol. 16, p. e1007794, 2020.
- [66] e. a. Yan, “Deep learning for brain tumor biomarker identification,” *Neuroinformatics*, vol. 17, pp. 345–356, 2019.
- [67] e. a. Li, “Integrative analysis of genomic and clinical data for glioma prognosis,” *Cancer Research*, vol. 78, pp. 1234–1245, 2018.
- [68] e. a. Chen, “Rna-seq based biomarker discovery for brain metastases using cnn,” *Scientific Reports*, vol. 10, p. 1234, 2020.
- [69] e. a. Bailey, “Liquid biopsies for brain tumor biomarker detection,” *Nature Reviews Cancer*, vol. 20, pp. 123–134, 2020.
- [70] C. Guinn and A. Habash, “Language analysis of speakers with dementia of the alzheimer’s type,” *2012 AAAI Fall Symposium Series*, pp. 8–13, 2012.
- [71] S. O. Orimaye, J. S.-M. Wong, and K. J. Golden, “Learning predictive linguistic features for alzheimer’s disease and related dementias using verbal utterances,” *2014 Association for Computational Linguistics*, pp. 78–87, 2014.
- [72] M. Yancheva and F. Rudzicz, “Vector-space topic models for detecting alzheimer’s disease,” *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, p. 2337–2346, 2016.
- [73] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, “Classification and regression trees,” *Wadsworth*, 1984.

- [74] L. Breiman, “Random forests,” *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [75] S. O. Orimaye, J. S.-M. Wong, and J. S. G. Fernandez, “Deep-deep neural network language models for predicting mild cognitive impairment,” *BAI@IJCAI*, p. 14–20, 2016.
- [76] B. Fischl, D. H. Salat, E. Busa, M. Albert, M. Dieterich, C. Haselgrove, A. Van Der Kouwe, R. Killiany, D. Kennedy, S. Klaveness, A. Montillo, N. Makris, B. Rosen, and A. M. Dale, “Whole brain segmentation: Automated labeling of neuroanatomical structures in the human brain,” *Neuron*, vol. 33, no. 3, pp. 341–355, 2002.
- [77] J. E. Iglesias and M. R. Sabuncu, “Multi-atlas segmentation of biomedical images: A survey,” *Medical Image Analysis*, vol. 24, no. 1, pp. 205–219, 2015.
- [78] M. Cabezas, A. Oliver, X. Lladó, J. Freixenet, and M. B. Cuadra, “A review of atlas-based segmentation for magnetic resonance brain images,” *Computer Methods and Programs in Biomedicine*, vol. 104, no. 3, pp. e158–e177, 2011.
- [79] T. Tong, R. Wolz, P. Coupé, J. V. Hajnal, and D. Rueckert, “Segmentation of mr images via discriminative dictionary learning and sparse coding: Application to hippocampus labeling,” *NeuroImage*, vol. 76, pp. 11–23, 2013.
- [80] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 234–241, 2015.
- [81] R. Roy, T. Chakraborti, and A. S. Chowdhury, “A deep learning-shape driven level set synergism for pulmonary nodule segmentation,” *Pattern Recognition Letters*, vol. 123, pp. 31–38, 5 2019.
- [82] S. Mukherjee, X. Huang, and R. R. Bhagalia, “Lung nodule segmentation using deep learned prior based graph cut,” in *IEEE ISBI*. IEEE, 2017, pp. 1205–1208.

- [83] A. De, M. Tiwari, E. Grisan, and A. S. Chowdhury, "A deep graph cut model for 3d brain tumor segmentation," in *EMBC*. IEEE, 2022, pp. 2105–2109.
- [84] T. Zhou, S. Ruan, and S. Canu, "A review: Deep learning for medical image segmentation using multi-modality fusion," *Array*, vol. 3, p. 100004, 2019.
- [85] A. De, R. Mhatre, M. Tiwari, and A. S. Chowdhury, "Brain tumor classification from radiology and histopathology using deep features and graph convolutional network," in *2022 26th International Conference on Pattern Recognition (ICPR)*, 2022, pp. 4420–4426.
- [86] A. De and A. S. Chowdhury, "Dti based alzheimer's disease classification with rank modulated fusion of cnns and random forest," *Expert Systems with Applications*, vol. 169, p. 114338, 2021.
- [87] A. De, M. Tiwari, and A. S. Chowdhury, "3d hippocampus segmentation using a hog based loss function with majority pooling," in *IEEE ICIP*. IEEE, 2023, pp. 2260–2264.
- [88] L. Ford and D. Fulkerson, *Flows in Networks*, ser. Princeton Landmarks in Mathematics and Physics. Princeton University Press, 2015. [Online]. Available: <https://books.google.co.in/books?id=fw7WCgAAQBAJ>
- [89] L. Breiman, "Random forests," *Machine learning*, vol. 45, pp. 5–32, 2001.
- [90] K. Hornik, M. Stinchcombe, H. White *et al.*, "Multilayer feedforward networks are universal approximators." *Neural networks*, vol. 2, no. 5, pp. 359–366, 1989.
- [91] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, pp. 533–536, 1986.
- [92] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *The Journal of physiology*, vol. 160, no. 1, p. 106, 1962.

- [93] K. Fukushima, “Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position,” *Biological cybernetics*, vol. 36, no. 4, pp. 193–202, 1980.
- [94] B. Shah and H. Bhavsar, “Time complexity in deep learning models,” *Procedia Computer Science*, vol. 215, pp. 202–210, 2022.
- [95] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, <http://www.deeplearningbook.org>.
- [96] S. Hasija, A. Peddaputha, M. B. Hemanth, and S. Sharma, “Video anomaly classification using densenet feature extractor,” in *Proceedings of International Conference on Computational Intelligence: ICCI 2021*. Springer, 2022, pp. 347–357.
- [97] R. Azad, E. K. Aghdam, A. Rauland, Y. Jia, A. H. Avval, A. Bozorgpour, S. Karimijafarbigloo, J. P. Cohen, E. Adeli, and D. Merhof, “Medical image segmentation review: The success of u-net,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.
- [98] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, 2012, pp. 1097–1105.
- [99] W. W. Zachary, “An information flow model for conflict and fission in small groups,” *Journal of anthropological research*, vol. 33, no. 4, pp. 452–473, 1977.
- [100] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, “Neural message passing for quantum chemistry,” in *International conference on machine learning*. PMLR, 2017, pp. 1263–1272.
- [101] P. Battaglia *et al.*, “Relational inductive biases, deep learning, and graph networks,” *arXiv*, 2018.

- [102] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, “The graph neural network model,” *IEEE transactions on neural networks*, vol. 20, no. 1, pp. 61–80, 2008.
- [103] D. Y. Qi, “Dr. yanjun qi, about deep2read,” 2019, accessed: 18 March 2025. [Online]. Available: <https://qdata.github.io/deep2Read>
- [104] S. Bauer, R. Wiest, L.-P. Nolte, and M. Reyes, “A survey of mri-based medical image analysis for brain tumor studies,” *Physics in Medicine & Biology*, vol. 58, no. 13, p. R97, 2013.
- [105] S. Osher and R. P. Fedkiw, “Level set methods: an overview and some recent results,” *Journal of Computational Physics*, vol. 169, no. 2, pp. 463–502, 2001.
- [106] Y. Y. Boykov and M.-P. Jolly, “Interactive graph cuts for optimal boundary & region segmentation of objects in nd images,” in *ICCV*, vol. 1. IEEE, 2001, pp. 105–112.
- [107] R. Roy, P. Banerjee, and A. S. Chowdhury, “A level set based unified framework for pulmonary nodule segmentation,” *IEEE Signal Processing Letters*, vol. 27, pp. 1465–1469, 2020.
- [108] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [109] Y. et al., “Edge-guided output adaptor: Highly efficient adaptation module for cross-vendor medical image segmentation,” *IEEE Signal Processing Letters*, vol. 26, no. 11, pp. 1593–1597, 2019.
- [110] T. et al., “Simultaneous segmentation and classification of mass region from mammograms using a mixed-supervision guided deep model,” *IEEE Signal Processing Letters*, vol. 27, pp. 196–200, 2020.
- [111] G. Wang and W. Li et al., “Interactive medical image segmentation using deep learning with image-specific fine tuning,” *IEEE Transactions on Medical Imaging*, vol. 37, no. 7, pp. 1562–1573, 2018.

- [112] Y. Hu and Y. Xia, “3d deep neural network-based brain tumor segmentation using multimodality magnetic resonance sequences,” in *International MICCAI Brainlesion Workshop*. Springer, 2017, pp. 423–434.
- [113] X. et al., “3d agse-vnet: an automatic brain tumor mri data segmentation framework,” *BMC Medical Imaging*, vol. 22, no. 1, pp. 1–18, 2022.
- [114] H. et al., “A deep multi-task learning framework for brain tumor segmentation,” *Frontiers in Oncology*, vol. 11, 2021.
- [115] S. Mukherjee, X. Huang, and R. R. Bhagalia, “Lung nodule segmentation using deep learned prior based graph cut,” in *ISBI*. IEEE, 2017, pp. 1205–1208.
- [116] R. Roy, T. Chakraborti, and A. S. Chowdhury, “A deep learning-shape driven level set synergism for pulmonary nodule segmentation,” *Pattern Recognition Letters*, vol. 123, pp. 31–38, 2019.
- [117] L. Liu, J. Cheng, Q. Quan, F. Wu, Y. Wang, and J. Wang, “A survey on u-shaped networks in medical image segmentations,” *Neurocomputing*, vol. 409, pp. 244 – 258, 2020.
- [118] Y. Boykov, O. Veksler, and R. Zabih, “Fast approximate energy minimization via graph cuts,” *IEEE TPAMI*, vol. 23, no. 11, pp. 1222–1239, 11 2001.
- [119] A. K. Rudra, M. Sen, A. S. Chowdhury, A. Elnakib, and A. El-Baz, “3d graph cut with new edge weights for cerebral white matter segmentation,” *Pattern Recognition Letters*, vol. 32, no. 7, pp. 941–947, 2011.
- [120] M. Abadi and A. A. et al., “TensorFlow: Large-scale machine learning on heterogeneous systems,” 2015, software available from tensorflow.org.
- [121] B. H. Menze, A. Jakab, and S. Bauer et al., “The multimodal brain tumor image segmentation benchmark (BRATS),” *IEEE Transactions on Medical Imaging*, vol. 34, no. 10, pp. 1993–2024, Oct. 2015.

- [122] S. Bakas and H. A. et al., “Advancing the cancer genome atlas glioma MRI collections with expert segmentation labels and radiomic features,” *Scientific Data*, vol. 4, no. 1, Sep. 2017.
- [123] B. C. Lowekamp, D. T. Chen, L. Ibáñez, and D. Blezek, “The design of SimpleITK,” *Frontiers in Neuroinformatics*, vol. 7, 2013.
- [124] C. R. Harris and et al., “Array programming with NumPy,” *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020. [Online]. Available: <https://doi.org/10.1038/s41586-020-2649-2>
- [125] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.
- [126] L. R. Dice, “Measures of the amount of ecologic association between species,” *Ecology*, vol. 26, no. 3, pp. 297–302, 1945.
- [127] P. Bourke, “Calculating the area and centroid of a polygon,” *Swinburne Univ. of Technology*, vol. 7, 1988.
- [128] V. Pedoia, S. Balbi, and E. Binaghi, “Fully automatic brain tumor segmentation by using competitive em and graph cut,” in *ICIAP*. Springer, 2015, pp. 568–578.
- [129] L. Grady, T. Schiwietz, S. Aharon, and R. Westermann, “Random walks for interactive organ segmentation in two and three dimensions: Implementation and validation,” in *MICCAI*. Springer, 2005, pp. 773–780.
- [130] S. Sasikanth and S. Suresh Kumar, “Glioma tumor detection in brain mri image using anfis-based normalized graph cut approach,” *International Journal of Imaging Systems and Technology*, vol. 28, no. 1, pp. 64–71, 2018.

- [131] K. Kamnitsas et al., “Efficient multi-scale 3d cnn with fully connected crf for accurate brain lesion segmentation,” *Medical Image Analysis*, vol. 36, pp. 61–78, 2017.
- [132] S. Alqazzaz, X. Sun, X. Yang, and L. Nokes, “Automated brain tumor segmentation on multi-modal mr image using segnet,” *Computational Visual Media*, vol. 5, no. 2, pp. 209–219, 2019.
- [133] I. et al., “No new-net,” in *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*. Cham: Springer International Publishing, 2019, pp. 234–244.
- [134] M. Havaei and A. Davy et al., “Brain tumor segmentation with deep neural networks,” *Medical Image Analysis*, vol. 35, pp. 18–31, 2017.
- [135] M. et al., “3d mri brain tumor segmentation using autoencoder regularization,” in *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*. Cham: Springer International Publishing, 2019, pp. 311–320.
- [136] H. Zhao, Y. Guo, and Y. Zheng, “A compound neural network for brain tumor segmentation,” in *ICIP*, 2019, pp. 1435–1439.
- [137] L. Sun, S. Zhang, H. Chen, and L. Luo, “Brain tumor segmentation and survival prediction using multimodal mri scans with deep learning,” *Frontiers in Neuroscience*, vol. 13, p. 810, 2019.
- [138] F. Bray et al., “Global cancer statistics 2018: Globocan estimates of incidence and mortality worldwide for 36 cancers in 185 countries,” *CA: A Cancer Journal for Clinicians*, vol. 68, no. 6, pp. 394–424, 2018.
- [139] F. G. Dhermain, P. Hau, H. Lanfermann, A. H. Jacobs, and M. J. van den Bent, “Advanced mri and pet imaging for assessment of treatment response in patients with gliomas,” *The Lancet Neurology*, vol. 9, no. 9, pp. 906–920, 2010.

- [140] V. Kumar, A. K. Abbas, and J. C. Aster, *Robbins basic pathology e-book*. Elsevier Health Sciences, 2017.
- [141] Z. Ye, R. L. Price, X. Liu, J. Lin, Q. Yang, P. Sun, A. T. Wu, L. Wang, R. H. Han, C. Song *et al.*, “Diffusion histology imaging combining diffusion basis spectrum imaging (dbsi) and machine learning improves detection and classification of glioblastoma pathology,” *Clinical Cancer Research*, vol. 26, no. 20, pp. 5388–5399, 2020.
- [142] T. Kurc *et al.*, “Segmentation and classification in digital pathology for glioma research: challenges and deep learning approaches,” *Frontiers in Neuroscience*, vol. 14, p. 27, 2020.
- [143] S. Sahayam, U. Jayaraman, and B. Teja, “Multi-class glioma classification from mri images using 3d convolutional neural networks,” in *International Conference on Computer Vision and Image Processing*. Springer, 2020, pp. 327–337.
- [144] H.-W. Chan, Y.-T. Weng, and T.-Y. Huang, “Automatic classification of brain tumor types with the mri scans and histopathology images,” in *International MICCAI Brainlesion Workshop*. Springer, 2019, pp. 353–359.
- [145] Y. Xue *et al.*, “Brain tumor classification with tumor segmentations and a dual path residual convolutional neural network from mri and pathology images,” in *International MICCAI Brainlesion Workshop*. Springer, 2019, pp. 360–367.
- [146] L. Pei *et al.*, “A hybrid convolutional neural network based-method for brain tumor classification using mmri and wsi,” in *International MICCAI Brainlesion Workshop*. Springer, 2020, pp. 487–496.
- [147] A. Hamidinekoo, T. Pieciak, M. Afzali, O. Akanyeti, and Y. Yuan, “Glioma classification using multimodal radiology and histology data,” in *International MICCAI Brainlesion Workshop*, 2020.

- [148] A. Bagari, A. Kumar, A. Kori, M. Khened, and G. Krishnamurthi, “A combined radio-histological approach for classification of low grade gliomas,” in *International MICCAI Brainlesion Workshop*. Springer, 2018, pp. 416–427.
- [149] A. Momeni, M. Thibault, and O. Gevaert, “Dropout-enabled ensemble learning for multi-scale biomedical data,” in *International MICCAI Brainlesion Workshop*. Springer, 2018, pp. 407–415.
- [150] X. Ma and F. Jia, “Brain tumor classification with multimodal mr and pathology images,” in *International MICCAI Brainlesion Workshop*. Springer, 2019, pp. 343–352.
- [151] L. Pei *et al.*, “Brain tumor classification using 3d convolutional neural network,” in *International MICCAI brainlesion workshop*. Springer, 2019, pp. 335–342.
- [152] N. J. Tustison, B. B. Avants, P. A. Cook, Y. Zheng, A. Egan, P. A. Yushkevich, and J. C. Gee, “N4itk: improved n3 bias correction,” *IEEE Trans. Med. Imag.*, vol. 29, no. 6, pp. 1310–1320, 2010.
- [153] Z. Yaniv, B. C. Lowekamp, H. J. Johnson, and R. Beare, “Simpleitk image-analysis notebooks: a collaborative environment for education and reproducible research,” *Journal of Digital Imaging*, vol. 31, no. 3, pp. 290–303, 2018.
- [154] A. De and A. S. Chowdhury, “Dti based alzheimer’s disease classification with rank modulated fusion of cnns and random forest,” *Expert Systems with Applications*, vol. 169, p. 114338, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0957417420310265>
- [155] S. Korolev, A. Safiullin, M. Belyaev, and Y. Dodonova, “Residual and Plain Convolutional Neural Networks for 3D Brain MRI Classification,” *2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017)*, pp. 835–838, 2017.

- [156] X. Wang *et al.*, “Weakly supervised deep learning for whole slide lung cancer image analysis,” *IEEE Trans. Cybernetics*, vol. 50, no. 9, pp. 3950–3962, 2019.
- [157] A. Goode, B. Gilbert, J. Harkes, D. Jukic, and M. Satyanarayanan, “Openslide: A vendor-neutral software foundation for digital pathology,” *J. Pathol. Inform.*, vol. 4, no. 1, p. 27, 2013.
- [158] G. Huang *et al.*, “Densely connected convolutional networks,” in *Proc. CVPR*, 2017, pp. 4700–4708.
- [159] T. Derr, Y. Ma, W. Fan, X. Liu, C. Aggarwal, and J. Tang, “Epidemic graph convolutional network,” in *Proceedings of the 13th International Conference on Web Search and Data Mining*. New York, USA: Association for Computing Machinery, 2020, p. 160–168. [Online]. Available: <https://doi.org/10.1145/3336191.3371807>
- [160] K. Farahani *et al.*, “Computational Precision Medicine Radiology- Pathology challenge on Brain Tumor Classification 2020,” Mar. 2020. [Online]. Available: <https://doi.org/10.5281/zenodo.3718894>
- [161] A. Paszke *et al.*, “PyTorch: an imperative style, high-performance deep learning library,” in *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc., 2019.
- [162] M. Fey and J. E. Lenssen, “Fast graph representation learning with PyTorch Geometric,” in *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.
- [163] X. Wang, S. Yang, and X. Wu, “Automatic glioma grading based on two-stage networks by integrating pathology and mri images,” in *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*. Springer International Publishing, 2021, pp. 455–464.
- [164] Q. T. Ostrom *et al.*, “The epidemiology of glioma in adults: a “state of the science” review,” *Neuro Oncol.*, vol. 16, no. 7, pp. 896–913, 2014.

- [165] W. L. Bi and R. Beroukhim, “Beating the odds: extreme long-term survival with glioblastoma,” *Neuro Oncol.*, vol. 16, no. 9, pp. 1159–1160, 2014.
- [166] A. BS, A. B, H. RS, V. V, A. Mahadevan, and H. J. Pandya, “Electromechanical characterization of human brain tissues: A potential biomarker for tumor delineation,” *IEEE Trans. Biomed. Engg.*, vol. 69, no. 11, pp. 3484–3493, 2022.
- [167] C. H. Suh, H. S. Kim, S. C. Jung, C. G. Choi, and S. J. Kim, “Imaging prediction of isocitrate dehydrogenase (idh) mutation in patients with glioma: a systemic review and meta-analysis,” *European radiology*, vol. 29, pp. 745–758, 2019.
- [168] P. C. Tripathi and S. Bag, “An attention-guided cnn framework for segmentation and grading of glioma using 3d mri scans,” *IEEE/ACM Trans. Comput. Biol. Bioinform.*, vol. 20, no. 3, pp. 1890–1904, 2023.
- [169] J. E. V. Vega and D. J. Brat, “Incorporating advances in molecular pathology into brain tumor diagnostics,” *Adv. Anat. Pathol.*, vol. 25, no. 3, pp. 143–171, 2018.
- [170] Z. Yi, L. Long, Y. Zeng, and Z. Liu, “Current advances and challenges in radiomics of brain tumors,” *Front. Oncol.*, vol. 11, p. 732196, 2021.
- [171] J. Shao, J. Ma, Q. Zhang, W. Li, and C. Wang, “Predicting gene mutation status via artificial intelligence technologies based on multimodal integration (mmi) to advance precision oncology,” in *Semin. Cancer. Biol.*, vol. 91. Elsevier, 2023, pp. 1–15.
- [172] A. De, R. Mhatre, M. Tiwari, and A. S. Chowdhury, “Brain tumor classification from radiology and histopathology using deep features and graph convolutional network,” in *Proc. ICPR*, 2022, pp. 4420–4426.
- [173] Y. Shen and J. Ke, “Sampling based tumor recognition in whole-slide histology image with deep learning approaches,” *IEEE/ACM Trans. Comput. Biol. Bioinform.*, vol. 19, no. 4, pp. 2431–2441, 2022.

- [174] L. Fu, Q. Peng, and L. Chai, “Predicting dna methylation states with hybrid information based deep-learning model,” *IEEE/ACM Trans. Comput. Biol. Bioinform.*, vol. 17, no. 5, pp. 1721–1728, 2020.
- [175] Y. Liu, A. Li, J. Liu, G. Meng, and M. Wang, “Tsd1pp: A novel two-stage deep learning framework for prognosis prediction based on whole slide histopathological images,” *IEEE/ACM Trans. Comput. Biol. Bioinform.*, vol. 19, no. 4, pp. 2523–2532, 2022.
- [176] R. B. Jenkins *et al.*, “A t (1; 19)(q10; p10) mediates the combined deletions of 1p and 19q and predicts a better prognosis of patients with oligodendroglioma,” *Cancer Res.*, vol. 66, no. 20, pp. 9852–9861, 2006.
- [177] Y. Li *et al.*, “Genotype prediction of atx mutation in lower-grade gliomas using an mri radiomics signature,” *Eur. J. Radiol.*, vol. 28, pp. 2960–2968, 2018.
- [178] N. Olympios, V. Gilard, F. Marguet, F. Clatot, F. Di Fiore, and M. Fontanilles, “Tert promoter alterations in glioblastoma: a systematic review,” *Cancers*, vol. 13, no. 5, p. 1147, 2021.
- [179] N. Saeed, S. Hardan, K. Abutalip, and M. Yaqub, “Is it possible to predict mgmt promoter methylation from brain tumor mri scans using deep learning models?” in *International Conference on Medical Imaging with Deep Learning*. PMLR, 2022, pp. 1005–1018.
- [180] Faghani *et al.*, “A comparison of three different deep learning-based models to predict the mgmt promoter methylation status in glioblastoma using brain mri,” *J. Digit. Imaging*, pp. 1–10, 2023.
- [181] B. Kocak *et al.*, “Radiogenomics of lower-grade gliomas: machine learning-based mri texture analysis for predicting 1p/19q codeletion status,” *European radiology*, vol. 30, pp. 877–886, 2020.

- [182] H. Zhou *et al.*, “Machine learning reveals multimodal mri patterns predictive of isocitrate dehydrogenase and 1p/19q status in diffuse low-and high-grade gliomas,” *Neuro Oncol.*, vol. 142, pp. 299–307, 2019.
- [183] T. Nishikawa *et al.*, “Easy-to-use machine learning system for the prediction of idh mutation and 1p/19q codeletion using mri images of adult-type diffuse gliomas,” *Brain Tumor Pathol.*, vol. 40, no. 2, pp. 85–92, 2023.
- [184] K. Masui *et al.*, “Elevated tert expression in tert-wildtype adult diffuse gliomas: Histological evaluation with a novel tert-specific antibody,” *BioMed. Research Int.*, vol. 2018, no. 1, p. 7945845, 2018.
- [185] B. Liechty, Z. Xu, Z. Zhang, C. Slocum, C. D. Bahadir, M. R. Sabuncu, and D. J. Pisapia, “Machine learning can aid in prediction of idh mutation from h&e-stained histology slides in infiltrating gliomas,” *Scientific Reports*, vol. 12, no. 1, p. 22623, 2022.
- [186] X. Liu *et al.*, “Multi-scale feature fusion for prediction of idh1 mutations in glioma histopathological images,” *Comput. Methods Programs Biomed.*, vol. 248, p. 108116, 2024.
- [187] D. Wang *et al.*, “Automated machine-learning framework integrating histopathological and radiological information for predicting idh1 mutation status in glioma,” *Front. Bioinform.*, vol. 1, p. 718697, 2021.
- [188] T. Wu, Q. Huang, Z. Liu, Y. Wang, and D. Lin, “Distribution-balanced loss for multi-label classification in long-tailed datasets,” in *Proc. ECCV*. Springer, 2020, pp. 162–178.
- [189] T. Ridnik, E. Ben-Baruch, N. Zamir, A. Noy, I. Friedman, M. Protter, and L. Zelnik-Manor, “Asymmetric loss for multi-label classification,” in *Proc. ICCV*, October 2021, pp. 82–91.
- [190] H. Yessou, G. Sumbul, and B. Demir, “A comparative study of deep learning loss functions for multi-label remote sensing image classification,” in *Proc. IEEE IGARSS*, 2020, pp. 1349–1352.

- [191] X. Xie *et al.*, “A survey on incorporating domain knowledge into deep learning for medical image analysis,” *Med. Image Anal.*, vol. 69, p. 101985, 2021.
- [192] E. Cerami *et al.*, “The cbio cancer genomics portal: an open platform for exploring multidimensional cancer genomics data,” *Cancer Discov.*, vol. 2, no. 5, pp. 401–404, 2012.
- [193] N. Dimitriou, O. Arandjelović, and P. D. Caie, “Deep learning for whole slide image analysis: an overview,” *Front. Med.*, vol. 6, p. 264, 2019.
- [194] A. Patil, H. Diwakar, J. Sawant, N. C. Kurian, S. Yadav, S. Rane, T. Bameta, and A. Sethi, “Efficient quality control of whole slide pathology images with human-in-the-loop training,” *J. Pathol. Inform.*, vol. 14, p. 100306, 2023.
- [195] B. Li, Y. Li, and K. W. Eliceiri, “Dual-stream multiple instance learning network for whole slide image classification with self-supervised contrastive learning,” in *Proc. CVPR*, 2021, pp. 14 318–14 328.
- [196] A. Ashtaiwi, “Optimal histopathological magnification factors for deep learning-based breast cancer prediction,” *Appl. Syst. Innov.*, vol. 5, no. 5, p. 87, 2022.
- [197] P.-T. De Boer, D. P. Kroese, S. Mannor, and R. Y. Rubinstein, “A tutorial on the cross-entropy method,” *Ann. Oper. Res.*, vol. 134, pp. 19–67, 2005.
- [198] H. Arita, Y. Narita, S. Fukushima, K. Tateishi, Y. Matsushita, A. Yoshida, Y. Miyakita, M. Ohno, V. P. Collins, N. Kawahara *et al.*, “Upregulating mutations in the tert promoter commonly occur in adult malignant gliomas and are strongly associated with total 1p19q loss,” *Acta neuropathologica*, vol. 126, pp. 267–276, 2013.
- [199] B. Nica, *A Brief Introduction to Spectral Graph Theory*, ser. EMS textbooks in mathematics. European Mathematical Society, 2018.
- [200] M. Tan and Q. Le, “Efficientnet: Rethinking model scaling for convolutional neural networks,” in *Proc. ICML*. PMLR, 2019, pp. 6105–6114.

- [201] D. Kumar and U. Batra, “Breast cancer histopathology image classification using soft voting classifier,” in *Proc. ICCIN*. Springer, 2021, pp. 619–631.
- [202] A. De and A. S. Chowdhury, “Dti based alzheimer’s disease classification with rank modulated fusion of cnns and random forest,” *Expert. Syst. Appl.*, vol. 169, p. 114338, 2021.
- [203] L. Scarpace *et al.*, “The cancer genome atlas glioblastoma multiforme collection (tcga-gbm),” *Canc. Imag. Arch.*, 2016.
- [204] N. Pedano, A. Flanders, L. Scarpace, T. Mikkelsen, J. Eschbacher, B. Hermes, V. Sisneros, J. Barnholtz-Sloan, and Q. Ostrom, “The cancer genome atlas low grade glioma collection (tcga-lgg)(version 3)[data set],” *Canc. Imag. Arch.*, 2016.
- [205] A. Riasatian *et al.*, “Fine-tuning and training of densenet for histopathology image representation using tcga diagnostic slides,” *Med. Image Anal.*, vol. 70, p. 102032, 2021.
- [206] S. Rajaraman, G. Zamzmi, and S. K. Antani, “Novel loss functions for ensemble-based medical image classification,” *Plos one*, vol. 16, no. 12, p. e0261307, 2021.
- [207] X. Wang *et al.*, “Prediction of brca gene mutation in breast cancer based on deep learning and histopathology images,” *Front. Genet.*, vol. 12, p. 661109, 2021.
- [208] N. Tomita *et al.*, “Predicting oncogene mutations of lung cancer using deep learning and histopathologic features on whole-slide images,” *Transl. Oncol.*, vol. 24, p. 101494, 2022.
- [209] E. Frank and M. Hall, “A simple approach to ordinal classification,” in *Proc. ECML*. Springer, 2001, pp. 145–156.

- [210] A. Martins and R. Astudillo, “From softmax to sparsemax: A sparse model of attention and multi-label classification,” in *Proc. ICML*. PMLR, 2016, pp. 1614–1623.
- [211] N. Mora *et al.*, “Comparison of mri sequences to predict atrx status using radiomics-based machine learning,” *Diagnostics*, vol. 13, no. 13, p. 2216, 2023.
- [212] J. Xiao, Y. Bai, A. Yuille, and Z. Zhou, “Delving into masked autoencoders for multi-label thorax disease classification,” in *Proc. WACV*, January 2023, pp. 3588–3600.
- [213] W. Weng, B. Wei, W. Ke, Y. Fan, J. Wang, and Y. Li, “Learning label-specific features with global and local label correlation for multi-label classification,” *Appl. Intell.*, vol. 53, no. 3, pp. 3017–3033, 2023.
- [214] C. Horbinski *et al.*, “The medical necessity of advanced molecular testing in the diagnosis and treatment of brain tumor patients,” *Neuro Oncol.*, vol. 21, no. 12, pp. 1498–1508, 2019.
- [215] O. Russakovsky *et al.*, “ImageNet Large Scale Visual Recognition Challenge,” *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.
- [216] E. Nichols, J. D. Steinmetz, S. E. Vollset, K. Fukutaki, J. Chalek, F. Abd-Allah, A. Abdoli, A. Abualhasan, E. Abu-Gharbieh, T. T. Akram *et al.*, “Estimation of the global prevalence of dementia in 2019 and forecasted prevalence in 2050: an analysis for the global burden of disease study 2019,” *The Lancet Public Health*, vol. 7, no. 2, pp. e105–e125, 2022.
- [217] 2019, Accessed: 2020-05-16. [Online]. Available: https://population.un.org/wpp/Publications/Files/WPP2019_Highlights.pdf
- [218] D. Le Bihan, J. F. Mangin, C. Poupon, C. A. Clark, S. Pappata, N. Molko, and H. Chabriat, “Diffusion tensor imaging: Concepts and applications,” *Journal of Magnetic Resonance Imaging*, vol. 13, no. 4, pp. 534–546, 2001.

- [219] F. Falahati, E. Westman, and A. Simmons, “Multivariate Data Analysis and Machine Learning in Alzheimer’s Disease with a Focus on Structural Magnetic Resonance Imaging,” *Journal of Alzheimer’s Disease*, vol. 41, pp. 685–708, 2014.
- [220] S. Rathore, M. Habes, M. A. Iftikhar, A. Shacklett, and C. Davatzikos, “A review on neuroimaging-based classification studies and associated feature extraction methods for Alzheimer’s disease and its prodromal stages.” *NeuroImage*, vol. 155, no. July, pp. 530–548, 2017.
- [221] X. W. Gao, R. Hui, and Z. Tian, “Classification of ct brain images based on deep learning networks,” *Computer Methods and Programs in Biomedicine*, vol. 138, pp. 49 – 56, 2017.
- [222] Z. N. K. Swati, Q. Zhao, M. Kabir, F. Ali, Z. Ali, S. Ahmed, and J. Lu, “Brain tumor classification for mr images using transfer learning and fine-tuning,” *Computerized Medical Imaging and Graphics*, vol. 75, pp. 34 – 46, 2019.
- [223] G. M. E. Elahi], S. Kalra, L. Zinman, A. Genge, L. Korngut, and Y.-H. Yang, “Texture classification of mr images of the brain in als using m-cohog: A multi-center study,” *Computerized Medical Imaging and Graphics*, vol. 79, p. 101659, 2020.
- [224] N. T. Duc, S. Ryu, M. N. I. Qureshi, M. Choi, K. H. Lee, and B. Lee, “3d-deep learning based automatic diagnosis of alzheimer’s disease with joint mmse prediction using resting-state fmri,” *Neuroinformatics*, vol. 18, no. 1, pp. 71–86, 2020.
- [225] S. Liu, Y. Song, W. Cai, S. Pujol, R. Kikinis, X. Wang, and D. Feng, “Multifold bayesian kernelization in alzheimer’s diagnosis,” in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2013, 16(Pt 2)*, K. Mori, I. Sakuma, Y. Sato, C. Barillot, and N. Navab, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 303–310.

- [226] M. Bucholz, X. Ding, H. Wang, D. H. Glass, H. Wang, G. Prasad, L. P. Maguire, A. J. Bjourson, P. L. McClean, S. Todd, D. P. Finn, and K. Wong-Lin, “A practical computerized decision support system for predicting the severity of alzheimer’s disease of an individual,” *Expert Systems with Applications*, vol. 130, pp. 157 – 171, 2019.
- [227] A. K. Ramaniharan, S. C. Manoharan, and R. Swaminathan, “Laplace beltrami eigen value based classification of normal and alzheimer mr images using parametric and non-parametric classifiers,” *Expert Systems with Applications*, vol. 59, pp. 208 – 216, 2016.
- [228] M. Raza, M. Awais, W. Ellahi, N. Aslam, H. Nguyen, and H. Le-Minh, “Diagnosis and monitoring of alzheimer’s patients using classical and deep learning techniques,” *Expert Systems with Applications*, vol. 136, pp. 353 – 364, 2019, Accessed: 2020-05-16.
- [229] D. Cheng, M. Liu, J. Fu, and Y. Wang, “Classification of mr brain images by combination of multi-cnns for ad diagnosis,” in *Proc.SPIE*, vol. 10420, jul 2017.
- [230] A. Lebedev *et al.*, “Random forest ensembles for detection and prediction of alzheimer’s disease with a good between-cohort robustness,” *NeuroImage: Clinical*, vol. 6, pp. 115 – 125, 2014.
- [231] D. L. Hall and J. Llinas, “An introduction to multisensor data fusion,” *Proceedings of the IEEE*, vol. 85, no. 1, pp. 6–23, 1997.
- [232] D. Castillo-Barnes *et al.*, “Autosomal dominantly inherited alzheimer disease: Analysis of genetic subgroups by machine learning,” *Information Fusion*, vol. 58, pp. 153 – 167, 2020.
- [233] T. Meng, X. Jing, Z. Yan, and W. Pedrycz, “A survey on machine learning for data fusion,” *Information Fusion*, vol. 57, pp. 115 – 129, 2020.
- [234] J.-H. Choi and J.-S. Lee, “Embracenet: A robust deep learning architecture for multimodal classification,” *Information Fusion*, vol. 51, pp. 259–270, 2019.

- [235] K. Kantarci *et al.*, “White-matter integrity on dti and the pathologic staging of alzheimer’s disease,” *Neurobiology of Aging*, vol. 56, pp. 172 – 179, 2017.
- [236] S. Y. Chun, K. C. Li, Y. Xuan, M. J. Xun, and W. Qin, “Diffusion tensor tractography in patients with cerebral tumors: A helpful technique for neurosurgical planning and postoperative assessment,” *European Journal of Radiology*, vol. 56, no. 2, pp. 197–204, 2005.
- [237] J. M. Soares, P. Marques, V. Alves, and N. Sousa, “A hitchhiker’s guide to diffusion tensor imaging,” *Frontiers in Neuroscience*, vol. 7, no. 7 MAR, pp. 1–14, 2013.
- [238] C. D. Mayo, E. L. Mazerolle, L. Ritchie, J. D. Fisk, and J. R. Gawryluk, “Longitudinal changes in microstructural white matter metrics in alzheimer’s disease,” *NeuroImage: Clinical*, vol. 13, pp. 330–338, 2017.
- [239] S. Pieper, M. Halle, and R. Kikinis, “3d slicer,” in *2004 2nd IEEE International Symposium on Biomedical Imaging: Nano to Macro (IEEE Cat No. 04EX821)*, 2004, pp. 632–635 Vol. 1.
- [240] A. Fedorov *et al.*, “3D Slicer as an image computing platform for the Quantitative Imaging Network,” *Magnetic resonance imaging*, vol. 30, no. 9, pp. 1323–1341, nov 2012.
- [241] K. Oishi, M. M. Mielke, M. Albert, C. G. Lyketsos, and S. Mori, “Dti analyses and clinical applications in alzheimer’s disease,” *Journal of Alzheimer’s Disease*, vol. 26, no. s3, pp. 287–296, 2011.
- [242] A. Abraham *et al.*, “Machine learning for neuroimaging with scikit-learn,” *Frontiers in Neuroinformatics*, vol. 8, p. 14, 2014.
- [243] F. Pedregosa *et al.*, “Scikit-learn: Machine learning in python,” *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011.
- [244] —, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

- [245] S. Nawar and A. M. Mouazen, “Comparison between random forests, artificial neural networks and gradient boosted machines methods of on-line vis-nir spectroscopy measurements of soil total nitrogen and total carbon,” *Sensors*, vol. 17, no. 10, 2017.
- [246] T. Chen and C. Guestrin, “Xgboost: A scalable tree boosting system,” in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.
- [247] F. Yang, H.-z. Wang, H. Mi, C.-d. Lin, and W.-w. Cai, “Using random forest for reliable classification and cost-sensitive learning for medical diagnosis,” *BMC bioinformatics*, vol. 10 Suppl 1, no. Suppl 1, pp. S22–S22, Jan 2009.
- [248] M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim, “Do we need hundreds of classifiers to solve real world classification problems?” *Journal of Machine Learning Research*, vol. 15, no. 90, pp. 3133–3181, 2014.
- [249] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “SMOTE: synthetic minority over-sampling technique,” *Journal of artificial intelligence research*, vol. 16, no. 2, pp. 321–357, 2002.
- [250] C. S. Greene, N. M. Penrod, J. Kiralis, and J. H. Moore, “Spatially uniform relief (surf) for computationally-efficient filtering of gene-gene interactions,” *BioData mining*, vol. 2, no. 1, p. 5, 2009.
- [251] C. R. Jack Jr. *et al.*, “The alzheimer’s disease neuroimaging initiative (adni): Mri methods,” *Journal of Magnetic Resonance Imaging*, vol. 27, no. 4, pp. 685–691, 2008.
- [252] X. Frazão and L. A. Alexandre, “Weighted convolutional neural network ensemble,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8827, pp. 674–681, 2014.

- [253] T. Raz, “The art of computer systems performance analysis: Techniques for experimental design, measurement, simulation, and modeling (raj jain),” *SIAM Review*, vol. 34, no. 3, pp. 518–519, 1992.
- [254] X. Bi, S. Li, B. Xiao, Y. Li, G. Wang, and X. Ma, “Computer aided alzheimer’s disease diagnosis by an unsupervised deep learning technology,” *Neurocomputing*, 2019.
- [255] T.-D. Vu, N.-H. Ho, H.-J. Yang, J. Kim, and H.-C. Song, “Non-white matter tissue extraction and deep convolutional neural network for alzheimer’s disease detection,” *Soft Computing*, vol. 22, no. 20, pp. 6825–6833, 2018.
- [256] D. Cheng, M. Liu, J. Fu, and Y. Wang, “Classification of mr brain images by combination of multi-cnns for ad diagnosis,” in *Ninth International Conference on Digital Image Processing (ICDIP 2017)*, vol. 10420. International Society for Optics and Photonics, 2017, p. 1042042.
- [257] K. Gunawardena, R. Rajapakse, and N. Kodikara, “Applying convolutional neural networks for pre-detection of alzheimer’s disease from structural mri data,” in *2017 24th International Conference on Mechatronics and Machine Vision in Practice (M2VIP)*. IEEE, 2017, pp. 1–7.
- [258] S. Liu, S. Liu, W. Cai, H. Che, S. Pujol, R. Kikinis, D. Feng, M. J. Fulham, and ADNI, “Multimodal Neuroimaging Feature Learning for Multiclass Diagnosis of Alzheimer’s Disease,” *IEEE Transactions on Biomedical Engineering*, vol. 62, no. 4, pp. 1132–1140, 2015.
- [259] B. Shi, Y. Chen, P. Zhang, C. D. Smith, and J. Liu, “Nonlinear feature transformation and deep fusion for alzheimer’s disease staging analysis,” *Pattern Recognition*, vol. 63, pp. 487 – 498, 2017.
- [260] B. Lei, S. Chen, D. Ni, and T. Wang, “Discriminative learning for alzheimer’s disease diagnosis via canonical correlation analysis and multimodal fusion,” *Frontiers in Aging Neuroscience*, vol. 8, p. 77, 2016.

- [261] N. Madusanka, H.-K. Choi, J.-H. So, and B.-K. Choi, “Alzheimer’s disease classification based on multi-feature fusion,” *Current Medical Imaging*, vol. 15, no. 2, pp. 161–169, 2019.
- [262] Z. Xiao, Y. Ding, T. Lan, C. Zhang, C. Luo, and Z. Qin, “Brain mr image classification for alzheimer’s disease diagnosis based on multifeature fusion,” *Computational and Mathematical Methods in Medicine*, vol. 2017, 2017.
- [263] C. La *et al.*, “Hippocampal ca1 subfield predicts episodic memory impairment in parkinson’s disease,” *NeuroImage: Clinical*, vol. 23, p. 101824, 2019.
- [264] Andersen *et al.*, *The hippocampus book*. Oxford university press, 2006.
- [265] E. E. Bron, M. Smits *et al.*, “Standardized evaluation of algorithms for computer-aided diagnosis of dementia based on structural mri: the cadde-mentia challenge,” *NeuroImage*, vol. 111, pp. 562–579, 2015.
- [266] E. Ghizoni, R. N. Matias *et al.*, “Clinical and imaging evaluation of transuncus selective amygdalohippocampectomy,” *World neurosurgery*, vol. 100, pp. 665–674, 2017.
- [267] M. L. Schlichting, D. Zeithamova, and A. R. Preston, “CA1 subfield contributions to memory integration and inference,” *Hippocampus*, vol. 24, no. 10, pp. 1248–1260, 10 2014.
- [268] M. J. Chadwick, H. M. Bonnici, and E. A. Maguire, “Ca3 size predicts the precision of memory recall,” *Proceedings of the National Academy of Sciences*, vol. 111, no. 29, pp. 10 720–10 725, 2014.
- [269] S. Baker *et al.*, “The human dentate gyrus plays a necessary role in discriminating new memories,” *Current Biology*, vol. 26, pp. 2629–2634, 2016.
- [270] N. L. Voets, B. C. Bernhardt, H. Kim, U. Yoon, and N. Bernasconi, “Increased temporolimbic cortical folding complexity in temporal lobe epilepsy,” *Neurology.*, vol. 76, no. 2, pp. 138–144, 1 2010.

- [271] K. H. Hobbs, P. Zhang, B. Shi, C. D. Smith, and J. Liu, “Quad-mesh based radial distance biomarkers for Alzheimer’s disease,” *Proceedings - International Symposium on Biomedical Imaging*, vol. 2016-June, pp. 19–23, 6 2016.
- [272] B. Fischl, “Freesurfer,” *Neuroimage*, vol. 62, no. 2, pp. 774–781, 2012.
- [273] W. Wang *et al.*, “A robust discriminative multi-atlas label fusion method for hippocampus segmentation from mr image,” *Computer Methods and Programs in Biomedicine*, vol. 208, p. 106197, 2021.
- [274] N. Safavian, S. A. H. Batouli, and M. A. Oghabian, “An automatic level set method for hippocampus segmentation in mr images,” *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, vol. 8, no. 4, pp. 400–410, 2020.
- [275] P. A. Yushkevich *et al.*, “Automated volumetry and regional thickness analysis of hippocampal subfields and medial temporal cortical structures in mild cognitive impairment,” *Human Brain Mapping*, vol. 36, no. 1, pp. 258–287, 1 2015.
- [276] J. E. Romero, P. Coupé, and J. V. Manjón, “HIPS: A new hippocampus subfield segmentation method,” *Neuroimage*, vol. 163, pp. 286–295, 12 2017.
- [277] M. Goubran *et al.*, “Hippocampal segmentation for brains with extensive atrophy using three-dimensional convolutional neural networks,” Wiley Online Library, Tech. Rep., 2020.
- [278] S. Suganyadevi, V. Seethalakshmi, and K. Balasamy, “A review on deep learning in medical image analysis,” *International Journal of Multimedia Information Retrieval*, vol. 11, no. 1, pp. 19–38, 2022.
- [279] Q. Huang, Y. Zhou, and L. Tao, “Dual-term loss function for shape-aware medical image segmentation,” in *ISBI*. IEEE, 2021, pp. 1798–1802.

- [280] I. Brusini *et al.*, “Shape information improves the cross-cohort performance of deep learning-based segmentation of the hippocampus,” *Frontiers in Neuroscience*, vol. 14, p. 15, 2020.
- [281] H. Tang *et al.*, “Shape-aware organ segmentation by predicting signed distance maps,” 2022, uS Patent 11,301,999.
- [282] J. V. Manjón, J. E. Romero, and P. Coupe, “A novel deep learning based hippocampus subfield segmentation method,” *Scientific Reports*, vol. 12, no. 1, 12 2022.
- [283] H. Zhu, F. Shi, L. Wang, S. C. Hung, M. H. Chen, S. Wang, W. Lin, and D. Shen, “Dilated dense U-net for infant hippocampus subfield segmentation,” *Frontiers in Neuroinformatics*, vol. 13, 4 2019.
- [284] Y. Shi, K. Cheng, and Z. Liu, “Hippocampal subfields segmentation in brain MR images using generative adversarial networks,” *BioMedical Engineering Online*, vol. 18, no. 1, 1 2019.
- [285] S. Suganyadevi, V. Seethalakshmi, and K. Balasamy, “A review on deep learning in medical image analysis,” *International Journal of Multimedia Information Retrieval 2021 11:1*, vol. 11, no. 1, pp. 19–38, 9 2021.
- [286] Z. Yang, X. Zhuang, V. Mishra, K. Sreenivasan, and D. Cordes, “CAST: A multi-scale convolutional neural network based automated hippocampal subfield segmentation toolbox,” *NeuroImage*, vol. 218, p. 116947, 9 2020.
- [287] X. Li *et al.*, “Syn_SegNet: A Joint Deep Neural Network for Ultrahigh-Field 7T MRI Synthesis and Hippocampal Subfield Segmentation in Routine 3T MRI,” *IEEE Journal of Biomedical and Health Informatics*, vol. 27, no. 10, pp. 4866–4877, 10 2023.
- [288] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *CVPR*, vol. 1. IEEE, 2005, pp. 886–893.

- [289] B. Bhattarai *et al.*, “Histogram of oriented gradients meet deep learning: A novel multi-task deep network for 2d surgical image semantic segmentation,” *Medical Image Analysis*, vol. 85, p. 102747, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1361841523000087>
- [290] A. Serag *et al.*, “Histograms of oriented 3d gradients for fully automated fetal brain localization and robust motion correction in 3 t magnetic resonance images,” *BioMed Research International*, vol. 2017, 2017.
- [291] Ö. Çiçek *et al.*, “3d u-net: learning dense volumetric segmentation from sparse annotation,” in *MICCAI*. Springer, 2016, pp. 424–432.
- [292] H. Wang, H. Zhang, and N. Ray, “Adaptive shape prior in graph cut image segmentation,” *Pattern Recognition*, vol. 46, no. 5, pp. 1409–1414, 5 2013.
- [293] J. Malcolm, Y. Rathi, and A. Tannenbaum, “Graph cut segmentation with nonlinear shape priors,” in *IEEE ICIP*, vol. 4. IEEE, 2007, pp. IV–365.
- [294] G. Li *et al.*, “Automatic liver segmentation based on shape constraints and deformable graph cut in ct images,” *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5315–5329, 12 2015.
- [295] G. Slabaugh and G. Unal, “Graph cuts segmentation using an elliptical shape prior,” in *IEEE ICIP*, vol. 2. IEEE, 2005, pp. II–1222.
- [296] A. M. Ali, A. A. Farag, and A. S. El-Baz, “Graph cuts framework for kidney segmentation with prior shape constraints,” in *MICCAI*. Springer, 2007, pp. 384–392.
- [297] O. Oktay *et al.*, “Attention u-net: Learning where to look for the pancreas,” in *Medical Imaging with Deep Learning*, 2018.
- [298] J. Kugelman *et al.*, “A comparison of deep learning u-net architectures for posterior segment oct retinal layer segmentation,” *Scientific Reports*, vol. 12, no. 1, p. 14888, 2022.

- [299] M. Solaiyappan, “Visualization pathways in biomedicine,” in *Handbook of medical imaging*. Elsevier, 2000, pp. 659–684.
- [300] P. Scovanner, S. Ali, and M. Shah, “A 3-dimensional sift descriptor and its application to action recognition,” in *Proceedings of the 15th ACM international conference on Multimedia*, 2007, pp. 357–360.
- [301] Y. Y. Boykov and M.-P. Jolly, “Interactive graph cuts for optimal boundary & region segmentation of objects in nd images,” in *ICCV*, vol. 1. IEEE, 2001, pp. 105–112.
- [302] Y. Boykov, O. Veksler, and R. Zabih, “Optimizing Multilabel MRFs Using Move-Making Algorithms,” *Markov Random Fields for Vision and Image Processing*, pp. 51–64, 12 2011.
- [303] A. Blake, P. Kohli, and C. Rother, *Markov random fields for vision and image processing*. MIT Press, 2011.
- [304] D. Freedman and T. Zhang, “Interactive graph cut based segmentation with shape priors,” in *IEEE CVPR*, vol. 1. IEEE, 2005, pp. 755–762.
- [305] J. Edmonds and R. M. Karp, “Theoretical improvements in algorithmic efficiency for network flow problems,” *Journal of the ACM (JACM)*, vol. 19, no. 2, pp. 248–264, 1972.
- [306] M. Boccardi *et al.*, “Training labels for hippocampal segmentation based on the eadc-adni harmonized hippocampal protocol,” *Alzheimer’s & Dementia*, vol. 11, no. 2, pp. 175–183, 2015.
- [307] K. Yoskovitz *et al.*, “Multi-contrast submillimetric 3 tesla hippocampal sub-field segmentation protocol and dataset,” *Scientific data*, vol. 2, no. 1, pp. 1–9, 2015.
- [308] C. R. Jack Jr *et al.*, “The alzheimer’s disease neuroimaging initiative (adni): Mri methods,” *Journal of Magnetic Resonance Imaging*, vol. 27, no. 4, pp. 685–691, 2008.

- [309] Y. Xu, F. Gao, T. Wu, K. M. Bennett, J. R. Charlton, and S. Sarkar, “U-net with optimal thresholding for small blob detection in medical images,” in *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*, 2019, pp. 1761–1767.
- [310] B. Thyreau, K. Sato, H. Fukuda, and Y. Taki, “Segmentation of the hippocampus by transferring algorithmic knowledge for large cohort processing,” *Medical Image Analysis*, vol. 43, pp. 214–228, 2018.
- [311] E. Balboni *et al.*, “The impact of transfer learning on 3d deep learning convolutional neural network segmentation of the hippocampus in mild cognitive impairment and alzheimer disease subjects,” *Human Brain Mapping*, vol. 43, no. 11, pp. 3427–3438, 2022.
- [312] D. Carmo *et al.*, “Hippocampus segmentation on epilepsy and alzheimer’s disease studies with multiple convolutional neural networks,” *Heliyon*, vol. 7, no. 2, p. e06226, 2021.
- [313] J. V. Manjón, J. E. Romero, and P. Coupe, “A novel deep learning based hippocampus subfield segmentation method,” *Scientific Reports*, vol. 12, no. 1, p. 1333, 2022.
- [314] K. Dwivedi, A. Rajpal, S. Rajpal, M. Agarwal, V. Kumar, and N. Kumar, “An explainable ai-driven biomarker discovery framework for non-small cell lung cancer classification,” *Comput. Methods Programs Biomed.*, vol. 153, p. 106544, 2023.
- [315] F. J. M. Shamrat, S. Akter, S. Azam, A. Karim, P. Ghosh, Z. Tasnim, K. M. Hasib, F. De Boer, and K. Ahmed, “Alzheimernet: An effective deep learning based proposition for alzheimer’s disease stages classification from functional brain changes in magnetic resonance images,” *IEEE Access*, vol. 11, pp. 16 376–16 395, 2023.

Arijit De
19/09/2024