

**A Hybrid Unsupervised Classification Technique for Automatic Brain
MRI Tumor Recognition**

by

Omar Irfan Khan

A Thesis

presented to

The University of Guelph

In partial fulfilment of requirements
for the degree of
Master of Science
in
Computer Science

Guelph, Ontario, Canada

© Omar Irfan Khan, January, 2020

ABSTRACT

A HYBRID UNSUPERVISED CLASSIFICATION TECHNIQUE FOR AUTOMATIC BRAIN MRI TUMOR RECOGNITION

Omar Irfan Khan
University of Guelph, 2020

Advisor:
Dr. Fangju Wang

Cancer is a prevalent disease with a rising incidence worldwide. The most common misdiagnosed type of cancers are brain tumors. Patients in early stages of brain tumors often present with vague symptoms. Due to the ambiguous nature of symptoms, diagnosis is difficult and the delay in diagnosis leads to a poor prognosis. In MRI images, brain tumors are hard to classify since they are embedded near normal tissues. This makes it hard for doctors to exactly extract tumors. Recognition of tumors in MRI images is tedious and time-consuming in most cases. To assist this process, several machine learning techniques have been developed. However, the techniques have limitations in practical diagnosis: Supervised techniques need a human operator to choose training data, while unsupervised techniques can only segment images without recognizing tumors. In this thesis research, we developed a hybrid technique for automatic brain MRI tumor recognition. This technique may overcome some limitations of the existing techniques as it does not require the involvement of a human operator and can effectively recognize tumors.

Acknowledgments

Foremost, I am grateful to my advisor Dr. Fangju Wang for his continuous patience, motivation and assistance during the writing of this thesis, and to Dr. Andrew Hamilton-Wright for his helpful comments and advice. Moreover, I am grateful to my friends Tamizhselvan V, Adefunke Akinola and Dillon Bourne for their words of encouragement, insightful comments and sticking around till the day ends. Finally, I would like to thank my family for everything they have done for me.

Table of Contents

Abstract	ii
Acknowledgements	iii
List of Tables	vi
List of Figures	vii
1 Introduction	1
1.1 Tumor Recognition in Brain MRI Images	1
1.2 Thesis Statement	2
1.3 Overview of Thesis	4
2 Related Work	5
2.1 Overview of Image Segmentation	5
2.2 MRI Images and Image Processing	7
2.2.1 MRI Imaging and Datasets	7
2.2.2 Image Preprocessing	9
2.2.3 Image Segmentation	10
2.2.4 Intensity-Based Segmentation	10
2.2.5 Atlas Based Segmentation	16
2.2.6 Deformable Model Based Segmentation	16
2.2.7 Hybrid Segmentation	18
2.2.8 Deep Learning Methods	18
2.3 Manual, Semi-Automatic and Automatic Segmentation	19
2.3.1 Manual Segmentation	20
2.3.2 Semi-Automatic Segmentation	21
2.3.3 Automatic Segmentation	23
3 Methodology and Implementation	25
3.1 An Overview	25
3.2 Preprocessing	25

3.3	Clustering	28
3.3.1	Comparison and Selection of Clustering Techniques	28
3.3.2	Implementation of the Clustering Technique	34
3.4	Thresholding	37
3.5	Evaluation and Analysis	40
4	Experimental Results and Analysis	41
4.1	Test Data and Experimental Procedures	41
4.2	Evaluation Criteria	42
4.3	Experimental Results and Analysis	51
4.3.1	Results of Identifying Tumors	51
4.3.2	Results of Processing Normal Images	52
4.3.3	Accuracy Analysis and Comparison	58
4.3.4	Comparison with Other Clustering Algorithms	62
5	Summary and Future Work	65
5.1	Summary of the Research	65
5.2	Future Work	65
	Bibliography	67
A	Appendix	76

List of Tables

2.1	MRI approaches along with their TR and TE	7
4.1	Precision, Recall and F1 Score of HGG and LGG	60
4.2	Time complexity of existing machine learning networks	60
4.3	Comparison against recent papers	63
A.1	Estimated cases for cancers [1]	77
A.2	Lifetime probability of developing cancer [1]	78

List of Figures

1.1	Axial (left), Sagittal (middle) and Coronal (right) views.	2
1.2	Raw tumor slice	3
2.1	An Agglomerative algorithm	12
2.2	Meanshift algorithm	14
2.3	Fuzzy C-Means algorithm	15
2.4	Different tumor identifications by four experts	22
3.1	The histograms of a brain image before preprocessing and after pre- processing	28
3.2	K-Means clustering algorithm.	30
3.3	DBSCAN algorithm	31
3.4	The HDBSCAN algorithm	33
3.5	Performance comparison of several clustering algorithms [2]	35
3.6	Performance comparison of high performance clustering algorithms [2]	36
3.7	Reconstructed LGG and HGG after clustering	39
3.8	Model Overview	40
4.1	Raw images of normal brain 1-10	43
4.2	Raw images of HGG 1-10	44
4.3	Raw images of LGG 1-10	45
4.4	Ground truth of HGG 1-10	46
4.5	Ground truth images of LGG 1-10	47
4.6	Confusion matrix of a HGG image	49
4.7	Confusion matrix of a LGG image	50
4.8	Clustered images of HGG 1-10	53
4.9	Clustered images of LGG 1-10	54
4.10	Thresholded images of HGG 1-10	55
4.11	Thresholded images of LGG 1-10	56
4.12	Output of normal brain after passing through our system 1-10	57
4.13	Ventricle highlighted	58

4.14 Initial clustering accuracy only	59
4.15 Accuracy of the overall system with different algorithms	60
4.16 Clustering comparison on HGG	61
4.17 Clustering comparing on LGG	61

Chapter 1

Introduction

1.1 Tumor Recognition in Brain MRI Images

Brain tumors are a leading cause of death in Canada. There are many types of brain tumors such as meningioma, pituitary adenoma, hemangioblastoma and glioma to name a few. This research is mainly focusing on identifying gliomas in MRI images since they occur most commonly as compared to others [3]. Gliomas can be further subdivided into high grade glioma (HGG) and low grade glioma (LGG).

Table A.1 illustrates the new cases of cancers in both sexes. Brain/Central Nervous System (CNS) has an estimate of 3,000 new cases in 2019. Table A.2 shows the lifetime probability of developing cancers in Canada excluding Quebec. Brain/CNS cancer can occur 1 in 157 in both sexes [1]. That is, the probability of developing Brain/CNS cancer is as high as 0.6%.

Computed Tomography (CT) was used widely for brain scans till magnetic resonance imaging (MRI) had some more recent advancements. MRI is far superior at displaying soft tissues as compared to CT and is being widely used for brain imaging to detect tumors. MRI shows a brain in four different modalities: T1, T1c, T2 and fluid attenuated inversion recovery (FLAIR). FLAIR reduces the contrast of normal tissues and lights up abnormalities such as a tumor. During an MRI process, several 2D slices of the brain are produced to represent the 3D brain. These slices can be taken in three views of the brain, axial (horizontal plane), sagittal (longitudinal plane) and coronal (frontal plane) which are shown in Figure 1.1.

MRI images can provide very useful information in diagnosing brain tumors.

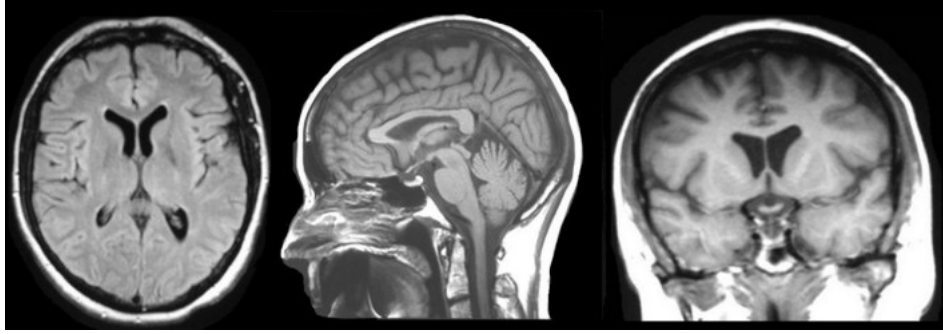


Figure 1.1: Axial (left), Sagittal (middle) and Coronal (right) views.

Unfortunately, the 2D slices produced are highly complex to interpret. They are riddled with noises and gray level values that are inconsistent or same between the lesion and normal areas. Gray level values are values that range from 0-255. 0 indicates that the area is completely black while, 255 are areas that are completely white. In Figure 1.2, we can see MRI images are prone to noises, which make accurate recognition of a tumor difficult. In addition, tumor segmentation in images can be challenging since a tumor may be spread over multiple slices. Therefore, it is crucial to identify the tumor accurately before the treatment can begin.

1.2 Thesis Statement

Brain MRI image interpretation is a laborious task for radiologists and doctors. The current approaches that are being widely used in hospitals include, purely manual interpretation and manual interpretation supported by digital image processing.

In a purely manual process, doctors conduct visual interpretation on each single slice and circulate the slices among them. Visual interpretation is a tedious task, which is labor intensive and time consuming [4]. A patient may miss the most suitable time for treatment because of delay in interpreting their brain images. An additional disadvantage of the purely manual interpretation is caused by subjectiveness. Limitation in human knowledge and experience often result in errors in brain image



Figure 1.2: Raw tumor slice

interpretation.

There are various digital image processing tools available online such as ITK-SNAP, 3D slicer and Dragonfly which help in 2D and 3D segmentation of anomalies in MRI images. In a manual interpretation supported by digital image processing, a software tool such as ITK-SNAP, 3D slicer, and Dragonfly, is used to detect tumor areas in images before human visual interpretation. In this approach, human examines only the images with suspected tumor areas, instead of all the images. This approach can largely improve the efficiency in diagnosing brain tumors and increase the chances that patients can be treated in time.

Currently, the software tools are mainly based on supervised learning to classify images into normal or tumorous. In a supervised learning, human involvement is required to choose labelled areas as training data for the software tool to learn. The process of choosing labelled data and training a classifier requires expertise and also

time consuming. As a result, interpreting MRI images in this way is still a bottleneck in brain tumor treatment.

My research is aimed at developing an unsupervised image processing technique to detect tumors in brain MRI images automatically. It is a classification technique to classify tumor areas from normal areas, instead of a segmentation technique to segment an image into different unlabelled areas. The technique is designed to provide accurate tumor information in a timely manner. The technique does not require human involvement, therefore processing can be conducted automatically and efficiency can be significantly improved. This technique integrates the techniques of hierarchical density based spatial clusterings of applications with noises (HDBSCAN), which is a density based clustering algorithm and thresholding to overcome the limitations of individual techniques. Thresholding is the most basic way to segment images. This is usually done using a grayscale image to create binary images for objects of interest.

1.3 Overview of Thesis

An introduction to the problem has been presented in Chapter 1 that indicates why there is a need to tackle brain tumors, how the current segmentation of brain tumors in MRI images is done and how an automated method can streamline the tumor segmentation process. A literature review is presented in Chapter 2 which sheds light on the current image processing techniques used, the various algorithms and different types of segmentation techniques that have been implemented to help solve this problem. Then in Chapter 3, the design of our automatic tumor recognition technique is described. The choices of technique components are discussed and justified. In Chapter 4, the experimental procedure and test data are described. Then experimental results are presented and analyzed. This thesis concludes in Chapter 5 with a summary of the work and major contribution, and how this work can be improved in the future.

Chapter 2

Related Work

2.1 Overview of Image Segmentation

Currently, several segmentation techniques have been developed and applied to digital medical images. Such techniques include thresholding, edge based, clustering, region of interest (ROI), watershed and machine learning techniques. Edge based segmentation uses edges to determine how to perform the segmentation. This is usually handled by edge based operators such as Canny. ROI based techniques usually require a creation of a mask to segment the ROI from the background. While, clustering techniques usually segment images based on their gray value. Similar gray values are grouped up in one cluster. Watershed is a classic image segmentation algorithm that has been widely used for medical image segmentation. It works on the idea that to separate areas in a grayscale image, the image is viewed as a surface and high intensities denote peaks while, low intensities denote valleys. Water (labels) are filled into valleys. As water rises the valleys may start to merge, therefore some barrier points have to be defined to prevent merging. These defined barriers result in the segmentation. Lastly, machine learning techniques can be divided into classical machine learning algorithms and deep learning algorithms. The latter are widely being used as compared to former due to their ability to determine features automatically. However, there is no robust technique agreed upon within the medical community.

Threshold based segmentation is based on certain pixel features i.e gray values. The pixel feature values are compared with segmentation thresholds to categorize

pixels. This approach is straightforward to implement.

Since boundary pixels are discontinuous, the pixel features tend to have evident distinction. An *edge-based segmentation* algorithm works to find the boundaries and specifies the directions of each boundary. After finding the boundary, pixels located in one boundary are placed in a segment. This process is repeated until all the pixels for each individual boundary found is placed in their individual segment. However, both these algorithms are vulnerable to noises and susceptible to inaccurate segmentation.

With the rise of machine learning, *neural networks* are being commonly used for image segmentation. Initially, neural networks require training on a dataset so the weights can be tuned between the nodes. After training is done, a neural net is given unseen data to see how the model can classify the image. One of the most common neural networks that have been widely used is the convolutional neural networks (CNNs) [5]. Although, one of the main disadvantages of neural nets is optimizing the parameters to attain a high functioning model. Moreover, image segmentation by neural networks are time consuming and resource intensive. This in turn, hampers the implementation and application.

Clustering algorithms are also used for image segmentation. Some of the commonly used clustering algorithms include fuzzy C-means, K-Means, and expectation maximization clustering. The K-Means algorithm is a globular clustering algorithm, which creates several globular clusters based on the cluster size. Image pixels are assigned to the nearest cluster centroid based on their distance. Fuzzy C-means clustering works on the idea of fuzzy set theory, which allows points to belong to multiple clusters. Lastly, the expectation maximization algorithm is a maximum likelihood method that does density estimation on the dataset. This is done by estimating the values that are missing and optimizing the other parameters at the same time. The main problem with this approach is that it requires a complete dataset otherwise it will be more computationally intensive. Also, expectation maximization is prone to noises which causes a problem especially with MRI images.

	TR (msec)	TE (msec)
T1-Weighted (Short TR and TE)	500	14
T2-Weighted(Long TR and TE)	4000	90
FLAIR (Very long TR and TE)	9000	114

Table 2.1: MRI approaches along with their TR and TE

2.2 MRI Images and Image Processing

2.2.1 MRI Imaging and Datasets

MRI imaging has changed how we visualize soft tissues. Previously, radiologists had to utilize a computed tomography scan (CT) to help in their visualization. While CT scans utilize X-rays to make a 3D model of the body, MRI scans use magnet and radio frequency waves to provide an elaborate view of internal biological structures in the form of slices. These slices are produced by RF pulses, which are varied to produce different types of weighted images. The main variables to vary are the repetition time (TR) and the time to echo (TE). The former denotes the amount of time between successive pulse sequences, while the latter is the delivery time of the RF pulse and the echo signal. The most commonly used MRI sequences are T1, T1c, T2 and FLAIR. T1 uses a very short TR and TE to obtain the image. T1c is just a T1 image with contrast applied to it. T2 uses a longer TR and TE time than T1 to obtain the image. Lastly, FLAIR uses an even longer TR and TE time to obtain the image. Table 2.1 shows the TR and TE time for the three common imaging modalities [6]. As the TR and TE increases, abnormalities tend to remain bright while healthy tissue is made dark. This in turn helps identify lesions. Furthermore, magnetic field strength plays a crucial role in determining how high the signal to noise ratio will be.

An important aspect in creating an automatic segmentation algorithm is to

choose the dataset to work on. There are various datasets available online but they can be categorized into two types:

1. Simulated Images
2. Real Images

Simulated images are usually rendered using online software tools or graphic tools to mimic actual MRI images. Simulated images cannot replace real brain images. Models that are trained on simulated images usually tend to have abysmal performance when dealing with real images. We use real images for my research. There are various image datasets available online:

1. The Cancer Imaging Archive (TCIA)
2. University of California Irvine (UCI) Data Repository
3. European Genome-Phenome Archive
4. Multimodal Brain Tumor Segmentation Challenge (BraTS)

TCIA is a website which contains various imaging datasets online. A problem with TCIA is that there are not many ground truth images available. The UCI data repository does not contain MRI brain slices as images rather has the patients information along with if they have a tumor. The tumor presence was denoted by a set of numbers and coordinates to determine where the tumor was located. Moreover, the dataset was released a long time ago. The European Genome-Phenome Archive has a dataset of 80 people with glioblastoma. However, One needs permission to access those files since they are not publicly available. The best choice is to use the BraTS dataset since it contains the ground truth and glioma images in various modalities. Moreover, a leaderboard for segmentation is also annually updated when they release a new dataset.

2.2.2 Image Preprocessing

Preprocessing plays a crucial role in clearing up ambiguities in images so that a image processing algorithm may function appropriately. Brain MRI images usually contain different matter resembling as the neck, skull and fat. These tissues have intensities that are similar to lesions. Therefore, it is vital to extract these features of interest before running a segmentation/classification algorithm. Due to the removal of features like skull and neck we end up with an image that only contains the brain along with the tumor (if any). Several methods have been developed to skull strip accurately by using the brain extraction tool (BET) which is an open source software system [7][8]. Furthermore, algorithms were also devised such as a watershed algorithm that was combined with deformable surface models to extract the brain from a T1-weighted image [9].

Another reason preprocessing is required is that MRI images may be riddled with noises and have a bias field also known as intensity inhomogeneity. Bias field and noises arise from discrepancies in the magnetic field. These discrepancies arise due to the settings at which the MRI machine is operated at. The settings can also be altered depending on the placement of coils in the machine or by a patients anatomy [10][11]. Since every patient has a different anatomy, same tissues may produce different intensities. Low pass filtering was proposed to help resolve bias field, but this approach eliminated all low frequencies including areas of interest. Other methods were proposed such as image entropy [12], registered template image [13] and increasing the high frequency points in an image [10]. Even after correcting the bias field some form of noises may still exist which can be addressed by using image denoising/smoothing filters. The goal of preprocessing is to produce images that have minimal noises and bias field, and are free from nonbrain tissues.

2.2.3 Image Segmentation

Image segmentation plays a key role in differentiating between healthy tissue and lesions. Brain tumor recognition proves to be a more challenging task for radiologists due to its complexity and anomalies induced under MRI [14]. Currently, most radiologists segment tumors manually using tools such as ITK-SNAP, 3D slicer. There have been several segmentation techniques proposed to help radiologists accurately identify tumors, such as:

1. Intensity-Based Segmentation
2. Atlas Based Segmentation
3. Deformable Model Based Segmentation
4. Hybrid Segmentation
5. Deep Learning Models

2.2.4 Intensity-Based Segmentation

Intensity represents the gray values present in an MRI image. There are several methods that differentiate pixels based on their intensities, mainly including thresholding, the methods developed by Otsu [15], triangle [16] and Huang [17], density based spatial clustering of applications with noises (DBSCAN), HDBSCAN, K-Means, Agglomerative, fuzzy C-means and Mean shift. The methods by Otsu, triangle and Huang fall under thresholding methods. The latter are clustering algorithms. Each clustering algorithm works differently. Some are centroid based and others are density based.

Hierarchical Clustering

Hierarchical clustering be divided into two types, agglomerative and divisive clustering. Agglomerative clustering is a bottom up approach, while divisive is a top

down approach. The main difference between them is how the clusters are formed. Agglomerative clustering usually starts with several small leafs and groups them into a bigger cluster, while divisive starts with one single cluster and splits it into smaller ones.

Agglomerative clustering works in the manner where one starts with a point in its own cluster and then for each cluster, some criterion value is used to merge with another cluster. This is done repeatedly until there is one cluster and a hierarchy is formed. A hierarchy in the shape of a dendrogram is usually formed. The dendrogram is cut and objects in the dendrogram are linked based on their similarity. There are several methods to calculate the similarity of clusters such as MIN, MAX and group average distances between centroids and wards method.

Ward's method is most commonly used as this method performs well if there are noises between clusters. Ward's method is based on a variance minimizing approach. Variance is measured by the index sum of squares. To obtain minimum variance in each cluster, the sum of squared differences is kept minimal between clusters. Ward's method which is also known as minimum variance method is often used as an alternative to single linkage clustering [18]. This is mainly due to the fact single linkage minimizes the distance between the closest pairs of clusters rather than all of the clusters. The method chooses the pair of clusters to merge based on the value from the objective function. The objective function in the case of ward's is the sum of squares. This method is often used along with agglomerative clustering since it produces compact, even sized clusters [19].

One of the problems with hierarchical clustering is that it does not indicate how many clusters there are or where to cut the dendrogram from. Moreover, even though ward deals with noises the problem still remains. It cannot effectively deal with noises since the method is biased towards globular clusters. Density based algorithms usually do not form globular clusters since they effectively weed out noise. While, centroid and hierarchical algorithms cannot eradicate noise and therefore, are included with

points of interest during cluster formation. Furthermore, the main problem with noises is that they are much harder to identify especially after the clusters have been established.

Agglomerative works similarly in some aspects to K-Means. K-Means is a centroid based clustering algorithm that requires parameters such as the cluster size prior to clustering. K-Means is the most commonly used clustering algorithm due to its great performance and time taken for the algorithm to execute. Although, agglomerative clustering is slightly better than K-Means, it can detect some noises but it has a high time complexity. Figure 2.1 shows the agglomerative cluster approach. The algorithm would be different if it was divisive since, it is a top down method.

Algorithm Agglomerative ($X[0\dots n-1]$, $n_clusters$, $affinity$, $linkage$)

Input: X is an array includes n objects,
 $n_clusters$ indicates the number of clusters to form,
 $affinity$ is used to calculate the distance between points,
 $linkage$ indicates the criterion to use between distances,

Output: Clusters

Initialize the clusters.

Compute the distance matrix

Find the two clusters with the closest distance

Put those two clusters into one cluster if the distance condition is satisfied

Repeat till all possible clusters agglomerated

Output clusters

End of algorithm

Figure 2.1: An Agglomerative algorithm

Meanshift

Meanshift is another centroid based algorithm, where one does not have to specify the number of clusters. Meanshift iteratively assigns each data point towards the closest cluster centroid. A kernel / probability density function is placed on each data point which can be controlled by the parameter bandwidth. The direction to

the closest cluster centroid is determined by where most of the points nearby are. In each iteration, the bandwidth also know as the window moves on to a set of data points. The mean for all those points in the window are calculated. Then the window is rearranged to accommodate the mean and the data points. This process is repeated until each point is assigned into a cluster and none of the data points are moving.

The kernel follows the approach of kernel density estimation which is a method that converges faster than binning and returns a continuous estimate for the density function. Binning is an approach most commonly used while making histograms. Essentially, binning groups a large number of data points into ranges of constant width. There are two conditions in choosing a kernel, one that the estimate should be normalized and the points should be associated with the symmetry of the models space. Two commonly used kernels abide by the following conditions, a flat/uniform kernel which is shown in Equation 2.1 and Gaussian kernel shown in Equation 2.2.

$$K(u) = \frac{1}{2} \begin{cases} 1 & -1 \leq |u| \leq 1 \\ 0 & else \end{cases} \quad (2.1)$$

$$K(u) = \frac{1}{(2\pi)^{d/2}} e^{-\frac{1}{2}|u|^2} \quad (2.2)$$

The main difference between K-Means and Meanshift is that, Meanshift does not need to specify the number of clusters. Meanshift is a very slow algorithm and using a flat or Gaussian kernel does not guarantee the center part of the kernel will be the densest part. In addition, Meanshift does not deal with any noises which are quite troublesome. Since the algorithm could not handle noises, preprocessing plays a key role for eliminating noises before using Meanshift.

Figure 2.2 shows the Meanshift algorithm. $N(x)$ indicates the neighbouring points, $m(x)$ indicates the meanshift equation which is basically the weighted average of the points that belong to x . The algorithm essentially, finds nearby points that are close to a data point and shifts the nearby points to the farther points. Convergence is reached when none of the points are moving and clusters have been formed.

Algorithm Meanshift (X[0...n-1],bandwidth)

Input: X is an array includes n objects,

Bandwidth is the window to decide which points to choose

Output: Clusters

Initialize clusters randomly

For each data point $x \in X$, find the neighbouring points $N(x)$

For each data point $x \in X$, calculate the mean shift $m(x)$ using equation:

$$m(x) = \frac{\sum_{x_i \in N(x)} K(x_i - x)x_i}{\sum_{x_i \in N(x)} K(x_i - x)}$$

For each data point $x \in X$, update $x \leftarrow m(x)$

Repeat for all data points for n.iterations or until convergence

Output clusters

End of algorithm

Figure 2.2: Meanshift algorithm

Fuzzy clustering

Fuzzy clustering algorithms are also a centroid based technique like K-Means, but data points in a fuzzy algorithm are not exclusive to one cluster and can be present in multiple clusters at the same time. Fuzzy C-means was developed by Dunn [20] and improved by Bezdek [21]. It works quite similarly to K-Means: we initially choose the number of clusters, assign coefficients randomly to each data point, repeat the calculations on the centroid for each cluster and computations for each data points coefficients until convergence.

The main difference from K-Means is the objective function of fuzzy C-means allows a data point to have multiple membership grades in different clusters, unlike K-Means, which has a strict objective function that allows only one cluster membership. Fuzzy C-Means performs well with overlapped data and is better than K-Means since

a single point belongs to different clusters based on its membership grade. However, Fuzzy C-Means still requires a parameter for the number of clusters to initialize like K-Means. Furthermore, fuzzy C-Means is still prone to noises since the algorithm cannot differentiate between noises and points of interest.

Algorithm Fuzzy C-Means ($X[0...n-1], n_clusters, metric$)

Input: X is an array includes n objects,
 $n_clusters$ indicates the number of clusters to form,
 $affinity$ is used to calculate the distance between points,
 $linkage$ indicates the criterion to use between distances
Output: Clusters

Randomly initialize clusters based on c

Repeat till constant values are obtained for the membership values:

Calculate the fuzzy membership for each data point using the equation A.1

Find the centroid using the formula in Equation A.2

Find the distance of each point from centroid

Update the membership values

Defuzzify the obtained membership values

Output clusters

End of algorithm

Figure 2.3: Fuzzy C-Means algorithm

Figure 2.3 shows the Fuzzy C-Means algorithm. V_j represents the j -th cluster center, m is the fuzziness index $m \in [1, \infty]$, c indicates the number of cluster centers, μ_{ij} is the membership of i -th data point to j -th cluster center and d_{ij} denotes the Euclidean distance between the i -th data point and the j -th cluster center. The fuzziness index or also know as a fuzzifier is used to tune out the noise.

2.2.5 Atlas Based Segmentation

People may have different shapes of organs. Medical images usually do not account for this. Atlases contain a population of images for a specific model with parameters that are learned from a training dataset [22][23]. Atlas based segmentation methods are able to segment regions without the need for any relation between regions and pixel intensities [24]. They segment unseen images by extrapolating from a labelled training dataset. Atlas segmentation can be divided into two types: parametric atlas and nonparametric atlas segmentation. Parametric atlas techniques combine all the training images into one atlas image [25], while nonparametric atlas methods use training images separately [26].

Atlas based methods require image registration, which is the process of matching the alignment of images. Image registration is required so the atlas image can be aligned to an unseen image. However, this is an exceedingly time intensive process since construction of the image is an iterative and complex process. Stefan et al. illustrated how atlas based segmentation is used on 2D slices to extract tumors [27]. Tumor growth was simulated using Markov random fields to show how effective the method is at isolating deformities. Moreover, atlas based segmentation can be improved when combined with other algorithms as Zhou and Bai demonstrated [28]. They proposed a framework which combined atlas registration and fuzzy connectedness. This gave atlas based techniques more variables to consider such as the various intensities during registration. Although, there is a problem with this method that MRI images with overlapping intensities cannot be handled with this system.

2.2.6 Deformable Model Based Segmentation

Deformable model based segmentation are models which consist of surfaces and curves. This segmentation technique is mostly used due its robustness and ease of use by non experts. These surface based techniques can be divided into three categories, level set and active contours. Deformable models can be summarized as methods that are a combination of physics, geometry and approximation theory.

Level set methods

Level set methods were introduced by Osher and Sethian [29][30], which are a way to demonstrate active contours. This method depends on position, time, the geometry of the interface and an energy function [31]. This approach builds a curve into a surface. The surface is manipulated while keeping the original curve in shape. Plus, this allows for better flexibility while viewing the object and works well as dimensionality increases. Kiran et al. improved the level set approach by having a new signed pressure function that stops contouring at weak or blurred edges [32]. Moreover, local statistics was utilized to differentiate between tumors and non tumor regions.

Active contours

Active contours are also known as snakes in two dimensions and active surfaces or active balloons in three dimensions which was originated by Kass et al. [33] for two dimensional spaces which was further generalized into three dimensional spaces by Terzopoulos et al. [34] and Cohen [35]. Active contours are curves that move within images to find object boundaries. This method is used to minimize the energy function from an external force, which resides in image data and internal force, which are curves or surfaces. Avina et al. illustrated a framework for brain tumor segmentation which involved a localized active contour with background intensity compensation [36]. This method allows handling images that have a high mean distance between an image in its background and foreground, which is usually the case with MRI images. Multiphase active contours also fall under active contours with a slight difference from the original, that multiple regions of an image can be identified with different characteristics. However, active contours, especially edge based active contours, may miss blurry boundaries and include outliers that do not pertain to the object. Moreover, they are sensitive to noises.

2.2.7 Hybrid Segmentation

There was a surge in the amount of algorithms being produced that could solve brain MRI segmentation problems [37][38]. However, people started seeing that using algorithms together can solve problems that are difficult by using a single algorithm. This in turn gave a rise to hybrid segmentation [39][40]. The former paper reported an active contour method along with deep belief network to differentiate tumors from non tumorous regions. While, the latter, combines self organizing maps with genetic algorithms. Genetic algorithm helps optimize what features are extracted and selected. Hybrid segmentation involves using two or more algorithms together to achieve the desired output.

Hybrid models tend to have a greater accuracy than other methods. Chaplot et al. used wavelets in combination with a support vector machine to classify abnormal and normal brain images [5]. They used T2 weighted, simulated images to carry out the experiments. A comparison was made against self organizing maps and support vector machines to see which classifier worked better. Akselrod-Ballin et al. demonstrated a hybrid technique that combined support vector machine with multi scale segmentation by weighted aggregation [41]. This method allowed for efficient feature extraction in terms of segregating brain regions such as white matter, cerebrospinal fluid and grey matter.

2.2.8 Deep Learning Methods

A subfield of machine learning called deep learning has gained significant attention in recent years. This is mainly due to an algorithm called convolutional neural network (CNN) which has successfully tackled different object recognition challenges [42] and medical image processing challenges [43]. CNNs tend to require a huge annotated dataset that acts as their training, validation and testing datasets. Furthermore, they require much computational power to execute. CNN can be divided into three layers, convolutional, pooling activation, and fully connected layers. Weights are used to adjust the network and are automatically learnt from training sets. CNNs learn

relationships between pixels of the input images and extract features that represent the image best using convolution and pooling. The first layer usually extracts the simpler features such as edges and the later layers extract more sophisticated features.

Urban et al. proposed a 3D CNN architecture for detection of MRI glioma segmentation [44]. Higher dimensionality was used to better represent the biological structures with more accuracy. Another approach by Havaei et al. involves a cascaded two pathway CNN architecture [45]. This technique involves using small sized patches along with bigger patches together. Furthermore, the output of the first pathway acts as an input for the second path way. In recent times patch based segmentation has greatly gained popularity and improved computational time of the network as a whole. Pereira et al. made several 3x3 sized filters in the convolutional layers which performed better than the usual large filter [46].

Besides CNNs, deep learning methods comprise of other neural networks such as multi layer perceptrons (MLP) and recurrent neural networks (RNN). They execute very well on complex, multivariate problems where decision trees or rule based systems generally do not perform well. However, a major drawback is that training samples may need to be manually annotated and training phase may be quite slow due to the various gradient learning techniques. Moreover, there is a chance bias may play a role during training. This may occur if the training data is not independent from the testing data. Furthermore, if the dataset is small, doing a split between training and test data may result in smaller sets for both training and testing. This in turn, reduces the reliability of the algorithm [47][48]. The most notable problem with deep learning algorithms is the difficulty in correctly tuning the parameters, gathering training samples and slow learning phase [49].

2.3 Manual, Semi-Automatic and Automatic Segmentation

Brain tumor segmentation can be divided into three types according to how much human involvement is required at each stage [50] [51]. These types are manual segmentation, semi-automatic segmentation and automatic segmentation.

2.3.1 Manual Segmentation

Manual segmentation involves manually highlighting the area where the brain tumor is located [50]. Moreover, since MRI scanners produce several slices of 2D brain images, a radiologist has to go manually into each slice and determine where the tumor is. Although, if the individual selecting or delineating is not well versed in identifying different brain regions, the segmentation yields poor results. Therefore, manually identifying tumors is highly operator dependent and time consuming. In recent years, applications such as ITK-SNAP, 3D slicer have emerged that help alleviate the time taken for segmentation. Needless to say, even though applications have been developed to make manual segmentation a bit faster, the fact still remains that MRI images are still subject to intra and inter rater variability [52].

Intra rater reliability is used to determine how consistent an individual is at measuring tumors, while inter rater reliability describes how various individuals are at identifying the tumor region. There are several factors that may affect intra rater reliability such as time, experience and the condition of the person. Some practitioners require more time for identifying the tumor as it may help the examiner focus more and not rush through the anomalies. Moreover, experience also comes into play here since examiners must have in depth understanding of the subject and have experience identifying various tumors. The markers condition is also an influencing factor since fatigue may lead to loss of focus and affect decision making. This metric is an important evaluating factor for manual segmentation since it eliminates external factors that can compromise the examiners evaluation. Also, this improves the examiners confidence in correctly identifying tumors and can increase the examiners competence when performed repeatedly.

In contrast, inter rater reliability is a metric used to determine the extent to which multiple examiners agree on the location of the tumor. The degree to which they will agree determines the reliability of the score. This metric allows us to impose level of objectivity, provides validation of evaluation results and increases confidence that examiners are following the proper guidelines in identifying a tumor.

Visser et al. compared inter rater agreement between experts and novices on gliomas [53]. They found that experts had a higher agreement as compared to novices during segmentation but, novices had poor performance at identifying tumors during progression and after postoperation. Meanwhile, experts also experienced trouble agreeing on the tumor region after postoperation and during its progression. Figure 2.4 shows an example taken from [54]. This example illustrates how inter rater variability differs between four experts on determining the area of the tumor. There are many notable differences between the experts when identifying the tumor.

Even if intra and inter rater variability vary between experts, manual segmentation is the only method currently to identify tumors correctly and is used as ground truth for semi-automatic segmentation and automatic segmentation methods. Development of methodologies such as semi-automatic and automatic segmentation have more advantages over manual segmentation. Manual segmentation is still more widely used due to the lack of development in reliable and accurate semi-automatic and automatic segmentation methods. Moreover, manual segmentation works well with distinguishing tissues as it combines human knowledge and expertise to reach a solution.

2.3.2 Semi-Automatic Segmentation

Semi-automatic segmentation needs human involvement to start a procedure, check the initial results or to provide better features to the algorithm. Currently, most of the research on brain MRI segmentation is focused on semi-automatic methods. Classic machine learning algorithms use hand crafted features and the features are inserted into algorithms like, support vector machine [55], random forest [56][57], fuzzy c-means [58], level set [59] and even particle swarm [60]. Among these algorithms, the most commonly used method is random forest due to its efficiency and ability to handle high dimensional data. Pinto et al. demonstrated a hierarchical extremely randomized forest based method for tumor segmentation [56]. This method achieved remarkable performance on the BRATS 2013 dataset. Since the rise of deep learning, many papers have published on CNNs of various architectures and some of them have

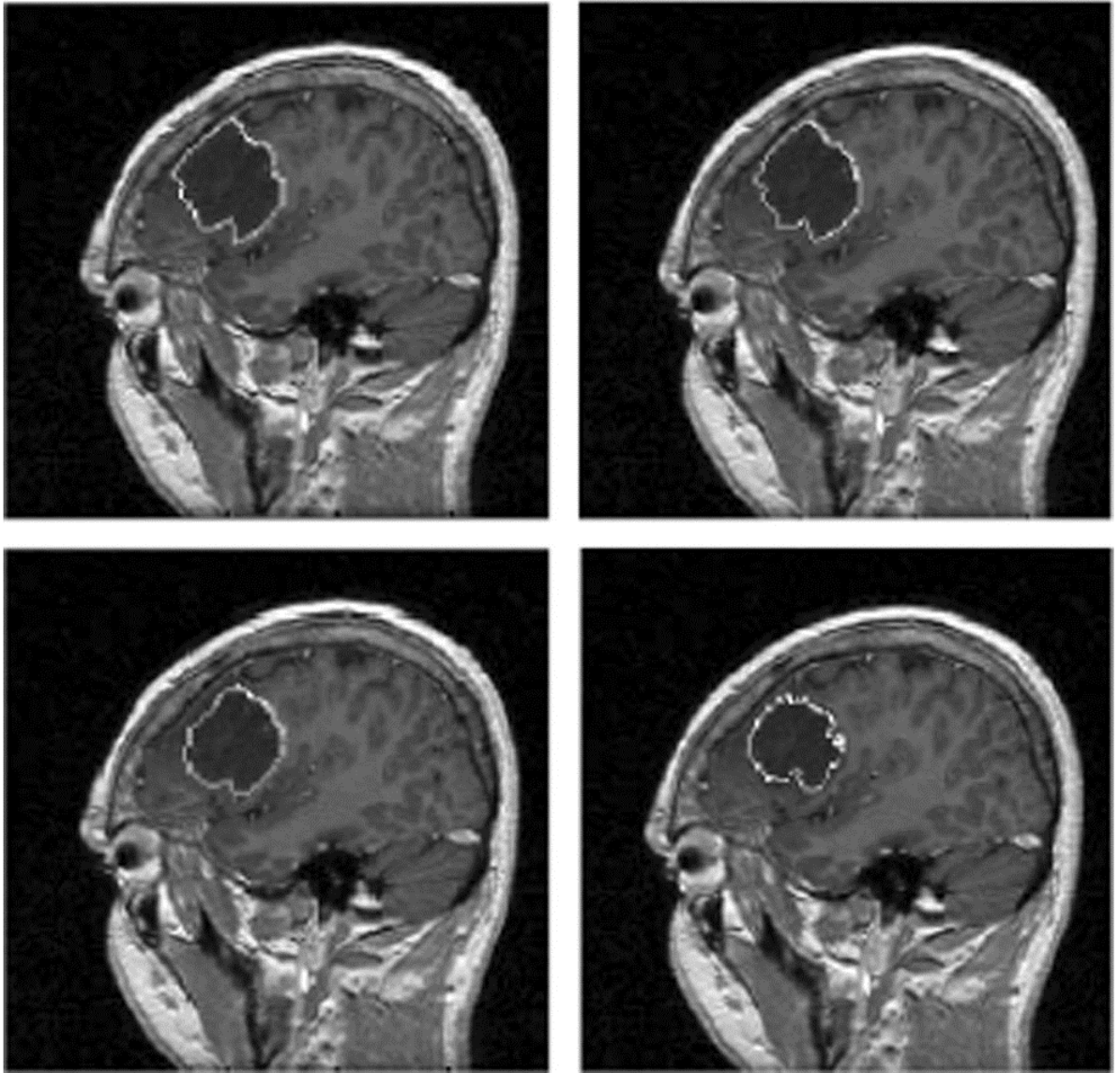


Figure 2.4: Different tumor identifications by four experts

achieved state of the art performance. There are some deep learning approaches which require human intervention for manual features. Some of the deep learning methods utilize supervised learning require the data to be labelled before hand. Therefore, they fall under semi-automatic segmentation. Pereira et al. proposed a simple CNN to segment gliomas. However, there was much human interaction involved since patches were made of the original image after preprocessing and then passed into the CNN [46].

2.3.3 Automatic Segmentation

Automatic segmentation allows the computer to determine location and segmentation of tumors without any human involvement. These methods usually incorporate human intelligence in the algorithms. Automatic segmentation has gained more traction, mainly because it tries to tackle the problem of how humans can effectively learn to segment and differentiate between brain tumor and normal regions. Although, development of automatic segmentation models are a challenging problem due to the fact that computers do not have strong visual processing abilities nor can they form a perfect link to perform the task effectively as compared to humans [61]. Nevertheless, this problem remains true for all pattern recognition, computer vision applications. However, computers do have the strength during brain tumor segmentation, that they can effectively display more information in 3D as compared to 2D which is, the normal format experts use to interpret where the tumor is located. Moreover, they can show the behaviour of different tissue types in different MRI modalities and the brain is well quantified structurally.

There are various papers published on segmentation mainly, since segmentation was a problematic issue for radiologists while dealing with brain tumors. Ali et al. discussed brain tumor segmentation methods based on MRI images [62]. They concluded that automatic segmentation is preferable to other segmentation methods. There were several versions of CNNs created most notably DeepMedic and U-net. Nonetheless, these tools also had some problems when learning from different datasets

which Shengcong et al. described, that using a dual force training strategy increases the quality of features learnt from deep models [63]. This was demonstrated on the BRATS 2015 and 2017 dataset using MLDeepMedic and U-Net. Moreover they utilized a semi-automatic MLP based method for post-processing to further enhance CNN prediction results while eliminating the need for manual summarization. A recent paper emphasized on how training and testing on different datasets may affect the segmentation accuracy and features obtained from it. Thus, reducing the models accuracy since each institution has different parameters tuned to obtain the MRI image [64]. There is no universal gold standard on the parameters used to obtain the MRI image. Nyoman et al. did tumor classification for gliomas, meningiomas and pituitary tumors which was compared against various architectures that have already been implemented. However, they were missing a color balancing step in preprocessing which in turn, led to obtaining lower accuracy for their model [65].

Grading of tumors is also an interesting issue since it involves classifying what type of tumor it is after segmentation. Sajjad et al. proposed a CNN to grade various types of gliomas, meningiomas and pituitary tumors by first segmenting the ROI using an input cascade CNN and then feeding it into their modified VGG 19 [66]. Anarki et al. tried a different hybrid approach attained brain tumor classification and grading via CNN while optimizing with genetic algorithm (GA) on different grades of glioma [67]. GA was used to evolve the best structure of CNN by choosing the proper parameters for the network. Two case studies were concluded, where one graded the various gliomas and the other assessed all the glioma, meningioma and pituitary images. Both these case studies obtained higher accuracy than previous studies.

For both papers [66][67] due to the lack of data, data augmentation was done to produce more images for each tumor. At the moment, fully automatic segmentation can process large amounts of MRI data faster without the need of human intervention. However, these methods have not gained a wide acceptance among practitioners due to the lack of transparency of how the model works and interpretability of the segmentation [68].

Chapter 3

Methodology and Implementation

3.1 An Overview

In our research, we developed a technique that can recognize brain tumors in MRI images automatically. It is a hybrid technique consisting of four stages to achieve automation of tumor area recognition:

1. Preprocessing: To reduce image biases and noises, we apply an N4 bias filter and a smoothing/denoising filter.
2. Clustering: To cluster the MRI scan, we use one of the clustering algorithms such as HDBSCAN.
3. Thresholding: The clustered image is passed into a thresholding filter to determine if the image contains a tumor.
4. Evaluation and Analysis: After thresholding, the image is passed into several external and internal cluster metrics.

The data used in the development are the BraTS 2018 dataset [69][70][71], which contains axial views of the brain in different modalities and brain tumor public dataset which was obtained from kaggle [72]. The dataset contains normal brain images for comparison.

3.2 Preprocessing

MRI images are sensitive to noises, biases and intensity inhomogeneity. A bias field is a low frequency signal that corrupts MRI images by blurring and distorting edges of

the image. This causes intensity based algorithms to perform poorly during segmentation and classification. To achieve high accuracy, we perform preprocessing before segmentation. This is often effective since if, preprocessing is not done algorithms tend to perform poorly due to noises and bias field which are always present in MRI images. After preprocessing we obtain an MRI image that has minimal noise and bias field.

The problems of bias field arise due to the magnetic settings, patients position and other factors. Although, some of these problems can be resolved by averaging multiple images directly in the scanner, this is not a common practice since this significantly increases the total image processing time. The issue of bias field can be resolved by using a Bias Field Correction algorithm. The algorithm works by taking a mask which specifies which pixels are used to estimate the bias field and suppress pixels closer to zero. Currently, there are two types of bias field correction algorithms N3 (nonparametric non uniformity normalization) and N4ITK.

N3 was proposed in 1998 and has been a fairly popular algorithm used for bias correction. However, it was noticed that N3 has an optimization problem which made convergence of points slower. The N3 algorithm does smoothing based on a basis spline (B-spline) approximation. B-spline is an expansion of spline which is a transformation of an image using curves. Splines work with piecewise polynomial function and are especially useful while solving problems that require smoothing or data interpolation.

The main difference between B-spline and spline is that, in the former, the output consists of the B-spline along with the variables while, the latter only produces the results. Nevertheless, there were various issues found with this approximation method. Firstly, the N3 approximation scheme violates the Schoenberg-Whitney conditions. Schoenberg-Whitney conditions apply for both curve fitting and surface fitting. The Schoenberg-Whitney theorem can be found in A.4. These conditions specify the placement of data and violation of them may cause an undetermined linear system leading to solution instability [73]. To combat this problem, N3 uses a weight regularization technique which needs to be tuned. The weight regularization

technique may lead to an unstable solution. In addition, another problem with N3 is that the algorithm is not robust to noise and has subpar results when dealing with noisy images.

N4ITK solves the problems of N3 by having a better B-spline approximation algorithm that conforms with the Schoenberg-Whitney conditions and does not rely on the least square method. This B-spline approximation method has a modified optimization scheme to provide an iterative update of the bias field, provides freedom to formation of different types of B-splines such as a cubic B-spline and performs optimally with multi resolution approximation. Moreover, rather than having B-spline approximation method done by the least squares approach which is susceptible to noise, the approximation in N4ITK is performed initially at the local level and then merged with the local solutions [74]. Therefore, the N4ITK algorithm is more efficient, robust and scalable to different resolutions [75].

In addition to the problem of bias fields, noises in images may severely reduce correctness of image segmentation. We address this problem by using a CurvatureFlowImageFilter. Other denoising algorithms/filters such as Non-Local Means Denoising, median filter, thresholding, Gaussian filter and Sobel remove areas of interest along with the noises or are not able to effectively eradicate all the noises. Plus, none of these denoising algorithms/filters can remove noise and smoothen to such an extent that the areas of interest are preserved [76].

The CurvatureFlowImageFilter is based upon an anisotropic diffusion method which is used for smoothing images while preserving edges. The filter attains this by running the curvature along contours, where they round the corners and reduce the Euclidean length of contours. The filter has two main parameters, which are the number of update iterations and the timestep between each update. The number of update iterations indicates how many iterations it requires, while the timestep ensures stability of the solution.

In the stage of preprocessing, the MRI image is first passed through a N4 Bias Field Correction filter to deal with the bias field. Next, the image from the N4 Bias Field Correction is passed through a CurvatureFlowImageFilter which implements a

curvature driven image denoising algorithm. This helps reduce noises and normalize the intensities. Figure 3.1 displays the raw tumor histogram and the tumor histogram after preprocessing. The noise points in the 60-100 cluster have been reduced and more reduction is seen as the gray levels get higher.

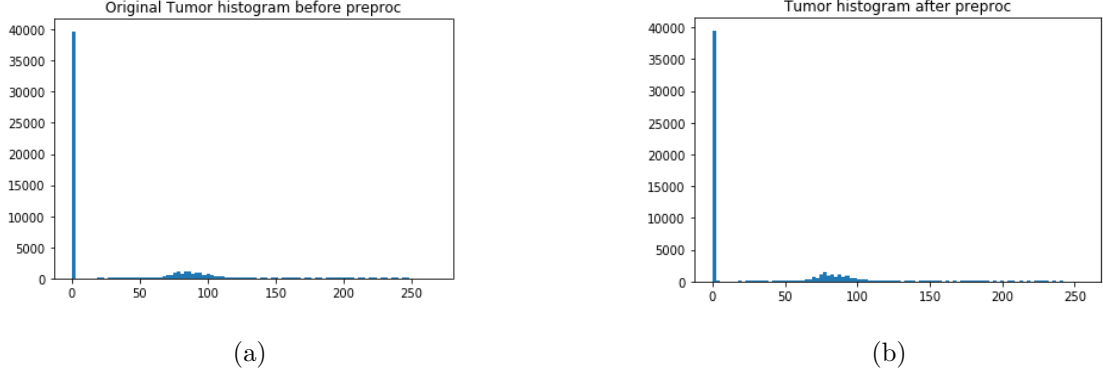


Figure 3.1: The histograms of a brain image before preprocessing and after preprocessing

3.3 Clustering

Clustering is a critical stage in automated tumor recognition in MRI images. Correct and accurate recognition of tumor areas relies on the quality of image segmentation. Also, suitable segmentation technique enables automation of tumor recognition. In this section, we discuss the suitability of three clustering algorithms: K-Means, DBSCAN and HDBSCAN in brain tumor segmentation. The pros and cons of each clustering algorithm for brain images are highlighted. Finally, we describe our selection of the most suitable clustering algorithm and its implementation in our method.

3.3.1 Comparison and Selection of Clustering Techniques

In designing our automatic tumor recognition method, we decided to incorporate an unsupervised clustering technique, because unsupervised clustering does not require human involvement. To choose the one most suitable for our purposes, we studied

and tested some commonly used unsupervised clustering algorithms. In the following, we compare K-Means, DBSCAN and HDBSCAN clustering algorithms, and justify our choice.

K-Means clustering

K-Means clustering is a centroid based algorithm that splits data into chunks while trying to converging to centroids of the chunks. Basically, the algorithm works by partitioning objects into K clusters where each object belongs to the cluster with the nearest mean. The object is assigned to a cluster if it is closer to that cluster's centroid than any other centroids. The distance is usually measured by an Euclidean function. In an iteration, after all objects are assigned to clusters, new means of the clusters are calculated. This process is repeated till mean values convergence or a predefined number of iterations is reached.

A problem with K-Means that one needs to specify how many clusters to expect which is not always feasible, especially if one does not know the data points well. Moreover, since K-Means assigns data points based on the distance between the point to the centroid, it does not account for the existences of outliers. Therefore, it is very prone to noises which is a big issue since MRI images are plagued with noises.

Figure 3.2 illustrates the K-Means clustering algorithm, where c indicates the number of clusters.

DBSCAN

DBSCAN works on the idea that dense regions belong to clusters and sparser regions belong to noise. It does this by using two parameters *epsilon* and *minpts*. Epsilon is a distance parameter that defines the radius to search for nearby neighbors, while *minpts* indicates the minimum number of points required to form a cluster. DBSCAN classifies data points into three types: core, border and noise points. A core point is a point that contains at least a minimum number of points within its *epsilon* radius. This is usually measured with a distance function such as Euclidean.

Algorithm K-Means ($X[0...n-1]$, $n_clusters=K$, n_init)

Input: X is an array includes n objects,

$n_clusters$ indicates the number of clusters to form,

n_init is the number of iterations the algorithm will run

Output: Clusters with data assigned according to their distance from the centroid

Choose the number of clusters(K) and obtain the data points

Place the centroids c_1, c_2, \dots, c_k randomly

for each data point x_i :

find the nearest centroid($c_1, c_2 \dots c_k$)

assign the point to that cluster

for each cluster $j = 1..k$

new centroid is mean of all points assigned to that cluster

Repeat for all unvisited points until convergence or until the iterations end

Output clusters

End of algorithm

Figure 3.2: K-Means clustering algorithm.

Border points on the other hand, are points that are within the *epsilon* radius of a core point but have less *minpts* within its own *epsilon* radius. Lastly, noise points are points that are neither a core or border point.

Each core point forms a cluster with the points that are reachable within its *epsilon* distance. The algorithm keeps assigning points to clusters until all of them have been visited once. It does not require all data points to be assigned. DBSCAN works very well as it does not require to specify the number of clusters and is resilient to noises.

A major advantage of DBSCAN is that outliers are not assigned to clusters, instead they are treated as noises. This feature avoids calculation of wrong centroids caused by noise points incorrectly assigned to clusters. A drawback of DBSCAN is choosing the *epsilon* value, since it is highly dependent on the user's familiarity with the data. Moreover, clusters of varying/similar densities do not get assigned properly. This is largely dependent on *epsilon* and the *minpts* parameter since these both combined give an estimate of the density required. Therefore, if the density is

not uniform in the image, DBSCAN will end up missing them, splitting them up or combining densities together while clustering based on the parameter choices.

Algorithm DBSCAN ($X[0\dots n-1]$, Epsilon, MinPts)

Input: X is an array includes n objects,

Epsilon specifies the radius to search for nearby neighbours,

MinPts implies the number of minimum points needed to form a cluster

Output: Clusters with data assigned according to their distance from the

$C = 0$

for each unvisited point P in X

mark P as visited

NeighborPts = regionQuery(P , epsilon)

find the nearest centroid($c_1, c_2 \dots c_k$)

if sizeof (NeighborPts) < MinPts

ignore P

else

$C =$ next cluster

expandCluster(P , NeighborPts, C , epsilon, MinPts)

expandCluster(P , NeighborPts, C , epsilon, MinPts):

add P to cluster C

for each point P in NeighborPts

if P is not visited

mark P as visited

NeighborPts = regionQuery(P , epsilon)

if sizeof(NeighborPts) \geq MinPts

NeighborPts = NeighborPts joined with NeighborPts

if P is not yet member of any cluster

add P to cluster C

regionQuery(P , eps):

return all points within P 's eps-neighborhood (including P)

End of algorithm

Figure 3.3: DBSCAN algorithm

Figure 3.3 shows the algorithm of DBSCAN, in which C represents the initial cluster counter. The function expandCluster() expands the cluster for all the points in the neighborhood. Function regionQuery() returns all the points in the neighborhood.

HDBSCAN

HDBSCAN is a fairly new algorithm which works in a similar way to DBSCAN except that it does not require an *epsilon* value and can determine which data points to consider as outliers. It works in a similar way to DBSCAN by starting to find the core distance of each point which is the distance between the point and its farthest neighbor. This is controlled by the *minpts* parameter. Moreover, a new metric is defined that is the *mutual reachability distance* which allows dense points with low core distance to remain the same distance from each other but sparser points are pushed away. This further reduces the probability that sparser points get added into clusters. Core distance is calculated based on the *min_cluster_size* to give an inexpensive measure of the density. It tries to find the smallest value of core distance that contains *K* neighbours. The clusters that are formed with DBSCAN and HDBSCAN are both formed in such a way that pixels with higher gray levels are in clusters with higher labels. Noises are usually addressed as -1. Since pixels in tumor areas usually have higher gray levels, both these algorithms identify tumor areas instead of just random gray values. This is the main reason we chose HDBSCAN.

The difference between HDBSCAN and DBSCAN arises on how to include points in a cluster. Usually DBSCAN takes *epsilon* into account which is an unintuitive parameter. HDBSCAN handles it in a different way by having a new parameter called *min_cluster_size* which determines whether points are falling out of a cluster or splitting to form two new cluster. This results in the exclusion of tiny off shoots and keeping bigger clusters as defined by the *min_cluster_size*.

HDBSCAN is especially suitable for varying densities which DBSCAN had a problem to deal with. Plus, HDBSCAN works better with higher dimensionality data and has a faster runtime as compared to DBSCAN. An MRI image contains several gray values which have varying densities. HDBSCAN starts by calculating the core distance of each point. Dense points usually have lower core distance as compared to low density points which have higher core distance. The mutual reachability metric allows denser points to stay closer while getting rid of outliers. These outliers are

placed in their own clusters away from gray values of interest. The gray values are then assigned according to their appropriate level to each cluster with their appropriate labels.

After obtaining the labels and clusters, the images are reconstructed a bit differently than centroid based algorithms. The initial array is compared with the labels and assigned as cluster points. While, the centers are calculated by taking the mean of cluster values. The cdist is calculated between the centers and the input array to determine if the distance between those points are accurate. Those points are then finally passed into centers to create the segmented image. Additionally, the algorithms were tuned with their respective hyperparameters using a validation dataset.

Algorithm HDBSCAN ($X[0\dots n-1]$, min_cluster_size)

Input: X is an array includes n objects,
min_cluster_size is the minimum cluster size
Output: Flat clusters having sizes not smaller than
min_cluster_size

Calculate core distance for each object in X
Calculate mutual reachability distance for each pair of objects
Create mutual reachability distance graph for X
Calculate minimum spanning tree (MST) from the graph
Convert the MST into cluster hierarchy which is a tree of single linkage
Condense the cluster tree top-down from the root till each node
becomes smaller than min_cluster_size
Extract flat clusters
Calculate stability values for nodes top-down.
For cluster C_i , its stability is calculated as

$$\sum_{x_j \in C_i} (\lambda_{\max}(x_j, C_i) - \lambda_{\min}(C_i))$$

$$= \sum_{x_j \in C_i} (1/\epsilon_{\max}(x_j, C_i) - 1/\epsilon_{\min}(C_i))$$

Select the nodes whose stability values are greater than
the sum of stability values of children
Output the selected nodes

End of algorithm

Figure 3.4: The HDBSCAN algorithm

Figure 3.4 is the algorithm of HDBSCAN, in which λ_{\min} is the minimum density level at which C_i exists, $\lambda_{\max}(x_j, c_i)$ is the density level beyond which x_j no longer

belongs to cluster C_i .

Comparison

The the main criteria to choose a clustering algorithm for our automatic method are dealing with outliers, forming clusters, and labelling clusters. As we have analyzed, HDBSCAN works better than DBSCAN and K-Means considering the criteria. There is another aspect to evaluate. That is the time efficiency of an algorithm to cluster data sets of large numbers of data points. In the following, we compare several clustering algorithm in terms of time efficiency.

Figure 3.5 shows the comparison with Sklearn K-Means, Scipy K-Means, Sklearn DBSCAN, HDBSCAN, Fastcluster Single Linkage, Scipy Single Linkage, Density Based Clustering (DeBaCl) Geom Tree, Sklearn Spectral, Sklearn Agglomerative and Sklearn Affinity propagation [2]. The graph can be divided into two classes, fast and slow implementations. From the graph we can see that agglomerative, affinity propagation, spectral and DeBaCl cannot handle data points from 10,000 effectively as the time taken to cluster significantly increases. Although, the algorithms that tend to performed well up to 30,000 data points were mainly, K-Means, DBSCAN, HDBSCAN and single linkage agglomerative clustering.

Figure 3.6 builds upon the fastest clustering algorithms with a larger number of data points [2]. Fastcluster starts to struggle here as the data points increase. Albeit, we can see the clear differences as HDBSCAN and DBSCAN start to struggle as well since they cannot achieve a time complexity of $O(n \log n)$. Since our image consists of points under 60,000, HDBSCAN is still the most viable option to choose due to its robust nature.

3.3.2 Implementation of the Clustering Technique

We implemented HDBSCAN by using the procedure described in the paper [77]. The algorithm has one parameter to tune which is the `min_cluster_size`. We tuned the parameter using ten validation images provided in the BraTs 18 dataset. After

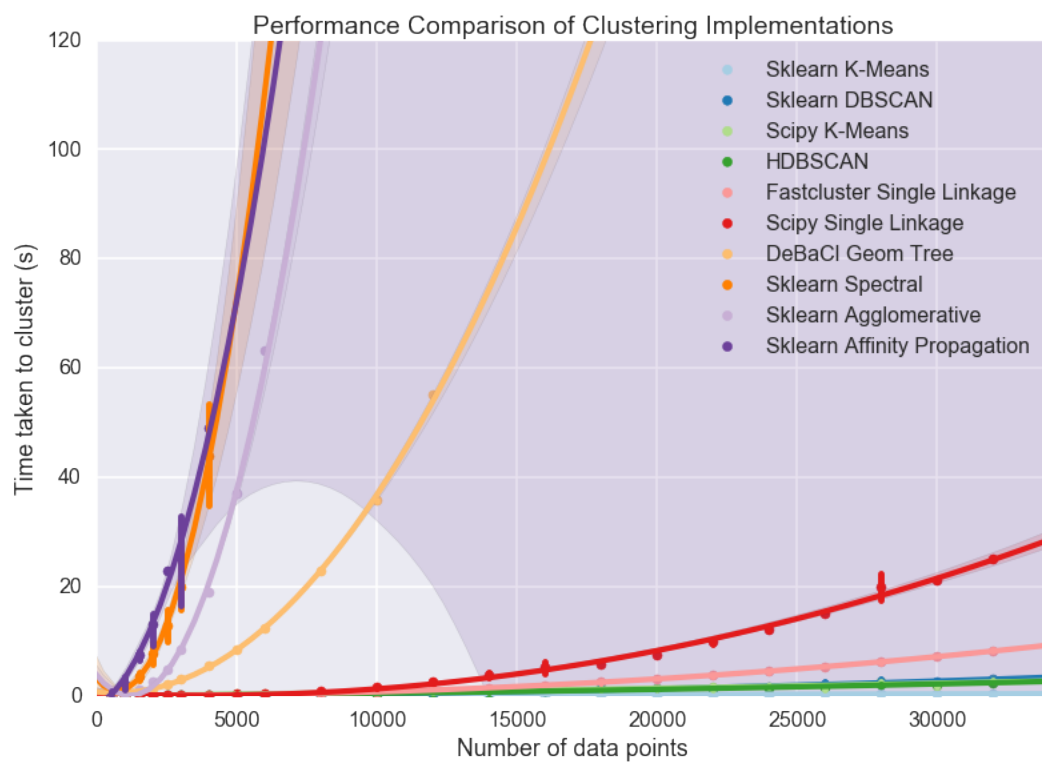


Figure 3.5: Performance comparison of several clustering algorithms [2]



Figure 3.6: Performance comparison of high performance clustering algorithms [2]

tuning, the optimal value found for `min_cluster_size` was thirty. The clustering technique in HDBSCAN assigns pixels with lower intensities to clusters with lower labels, and pixels with higher intensities in clusters with higher labels. Moreover, sparser points are excluded during the construction of the tree due to the mutual reachability distance which is shown in Equation A.3. Thus, those points are assigned a value of -1 and are discarded during the reconstruction of the image obtained from HDBSCAN.

3.4 Thresholding

The result of HDBSCAN on an MRI brain image is a segmented multi-value image, in which a tumor area (if any) is one of the segmented areas. To identify tumor area(s) in the image, we incorporate a stage of thresholding to generate a binary image. If there is a tumor area, it is highlighted by a higher intensity in the binary image, otherwise, the image has really low intensity assigned. A threshold is used to classify pixels into two classes.

There were several methods to choose from ranging from machine learning methods such as a CNN to algorithms such as fuzzy c means. Since we wanted a fully automated system that had relatively low processing time and did not rely on prior knowledge. We decided to go with thresholding since it meets most of these criteria.

A task in the thresholding is to choose a threshold to classify pixels. We have automated that process which will be elaborated in a later subsection. Thresholding is the simplest way to segment objects from the background. The basic thresholding method requires a predefined gray value which is usually chosen from the histogram of the image. The conventional method requires a human to interpret the histogram distribution and decide what gray value to pass for segmentation. Based on the predefined gray value, any value that falls under the predefined gray value is omitted from the result as the gray value is set to zero.

To help automate the generation of a binary image, we developed an algorithm to choose threshold values by examining histograms. This was done by analyzing histograms of general tumor images and scanning histogram of the image segmented by

HDBSCAN. In brain MRI images, tumors usually have high intensities. We observed that tumor areas usually have gray values between 130-180. As mentioned before, we may control the labelling of segments in the result of HDBSCAN. This enables a program to scan the histograms of both the raw image and segmented image to find a threshold that can highlight tumor area(s) in the segmented image.

We conducted experiments to determine where to integrate thresholding, before, during or after clustering, without sacrificing accuracy and time efficiency. When thresholding was done prior to clustering, thresholding does not provide adequate results for every gray level, the same threshold value is applied. If the gray value is smaller than the threshold value, the gray value is set to zero i.e. excluded from the output. This is a bit of an issue since this will set all the noise values plus some ROI gray values to zero. Thus, a significant amount of data is lost via thresholding before. Moreover, if one were to pass the resulting image into any one of the clustering algorithms, it produces the same image with zero noise since the image has already been filtered.

Figure 3.7 shows if thresholding was done before clustering on both LGG and HGG. The image was reconstructed after clustering with HDBSCAN. Since it is not a centroid based algorithm reconstructing the image was quite different from centroid based algorithms. Some of the data was lost and noise points were added. The noise points can mainly be indicated with some of the white points around the initial lesion. While, doing thresholding during and after clustering the results produced were the same in terms of accuracy and time elapsed. Therefore, we decided to perform thresholding after clustering.

After obtaining the images from the clustering algorithms, they are passed through a thresholding filter to further extract the brain tumor. Since our system is entirely automated, the value for thresholding is decided based upon the histogram. Tumors have higher gray levels, therefore, higher values are usually chosen for the threshold to take place at.

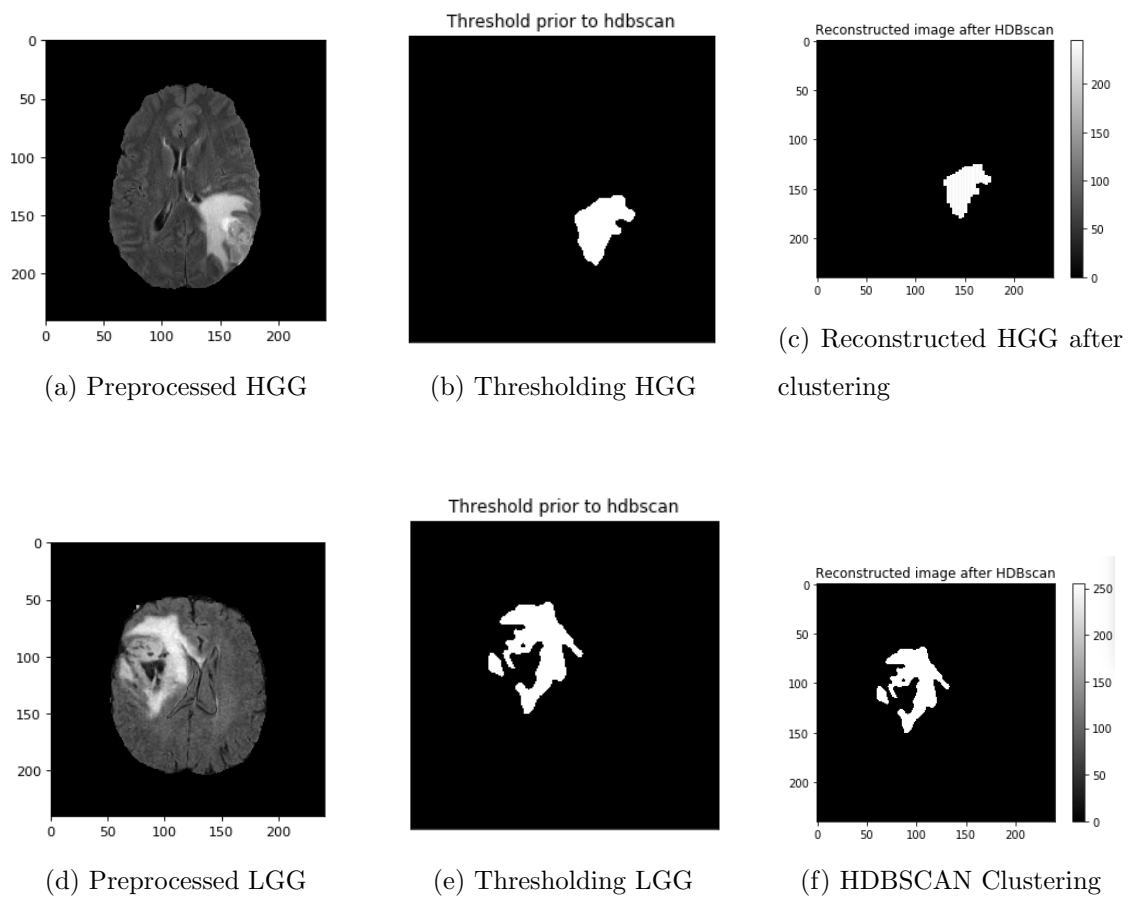


Figure 3.7: Reconstructed LGG and HGG after clustering

3.5 Evaluation and Analysis

After obtaining the results, statistics such as accuracy, time elapsed, true positive ratio, false positive ratio, F1 score, precision, recall are carried out. Moreover, some internal indices measures such as Calinski harabasz, Silhouette and Davies-Bouldin score were also carried out. An additional comparison is also set up to validate our results initially, to see if we obtained the tumor appropriately.

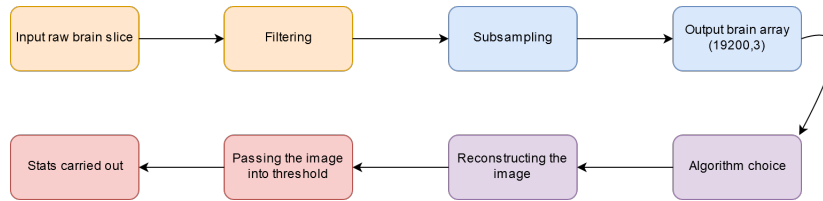


Figure 3.8: Model Overview

Figure 3.8 summarizes the entire process of brain tumor recognition. A raw slice (2D) from an image containing several slices is chosen and passed through a bias field correction and a smoothing filter. The image array is reformatted into another shape to provide better performance and normalized to have a consistent input array for all images. The resulting array is passed into one of the clustering algorithms to for clustering. After clustering the image is reconstructed by using the centroids/core points and labels. Moreover, the clustered array is then passed into thresholding where the threshold value gets automatically chosen based on the histogram values. Lastly, statistics are carried out to determine how the predicted result compares with the ground truth.

Chapter 4

Experimental Results and Analysis

4.1 Test Data and Experimental Procedures

The BraTS dataset contains 351 images combined from other datasets [78][79]. The dataset contains two folders for training and validation. We randomly selected twenty tumor slices to test our technique. Ten of them were HGG and ten were LGG. Moreover, ten random normal slices were also chosen from the brain tumor public dataset, to test how our technique deals with areas that have high gray levels but are not tumors. The higher gray levels may arise due to using different modalities of MRI to find the tumor or may occur due to contrast.

Figure 4.1 shows the normal brain images in different modalities. Figure 4.2 shows the raw HGG images, while Figure 4.3 shows the raw LGG images. The images show that it is not always obvious to accurately identify, locate, and delimit the tumor regions due to the presence of non tumor regions i.e. normal brain tissues and other factors. The ground truth images of the HGG are in Figure 4.4 and the ground truth images of LGG are in Figure 4.5. The ground truth images contain some white and gray areas. Gray areas denote that the tumors have undergone necrosis or has dead cells within them, while the white areas shows the active parts of the tumors.

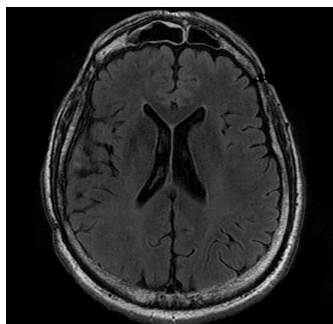
We tested our HDBSCAN+thresholding technique on all the 30 normal and tumor images. For each image, the technique generated a binary image as tumor detection result. We evaluated each result. In addition, comparisons are made between our technique (HDBSCAN+thresholding) and K-Means, Agglomerative, Meanshift and DBSCAN. The algorithms are compared mainly in terms of accuracy and efficiency. In the experiments, we tested the techniques on the 20 tumor slices separately, and

then evaluated the overall performance of each technique. Hyperparameter tuning was done on all the respective algorithms before clustering. The tuning was done by utilizing ten random images from the validation images provided. K-Means has a parameter called `n_clusters` which indicates how many clusters do you want to create and `n_init` which indicates how many iterations do you want it to run for. Normally, the iterations are set as default to 30. While, `n_clusters` was tuned. Agglomerative has a parameter called `n_clusters` and `linkage` which are two main features to be tuned. `n_clusters` indicates the amount of clusters to start with while, `linkage` criteria clarifies what strategy to use to merge the clusters. Linkage can either be Ward, Maximum or complete linkage, average linkage or single linkage. We found that Ward gave us a better result as compared to the rest. Meanshift depends on a parameter called `bandwidth` which determines the size of the window in which you want to calculate the mean. The easiest way to determine this is by using a module provided in scikit-learn called `estimate_bandwidth()` [80]. In DBSCAN, the parameter `epsilon` and `minpts` needs to be tuned. The former indicates the location of the cut off point and acts as the radius for core points. Lastly, HDBSCAN has the parameter `min_clust_size` which defines how many points are needed to form the cluster or to determine where to cut the tree based on the points falling off. All the parameters listed above were tuned on ten random validation images and were evaluated using twenty images. In addition, we tested our technique on normal brain images to see its behavior in differentiating between tumor regions and normal brain regions that are more likely contrast enhanced.

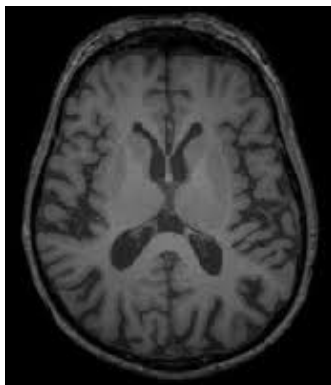
4.2 Evaluation Criteria

There are several evaluation criteria to determine if the tumor recognition was successful. The most commonly used criteria are accuracy, precision, recall and efficiency.

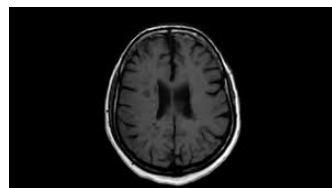
Accuracy measures the number of pixels correctly classified divided by the total number of pixels classified multiplied by 100. Accuracy can be calculated by using a confusion matrix (also called error matrix). In a confusion matrix, the cell at row



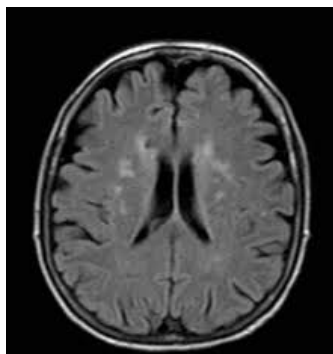
(a) normal brain 1



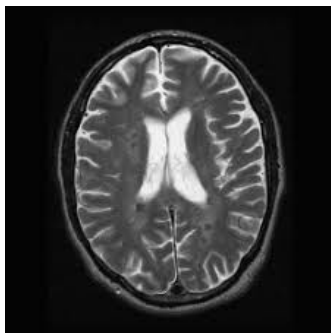
(b) normal brain 2



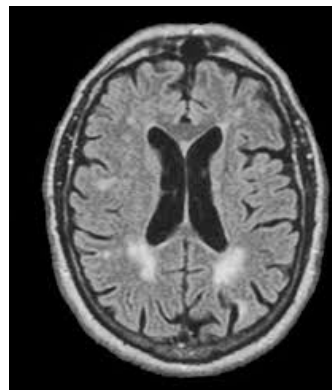
(c) normal brain 3



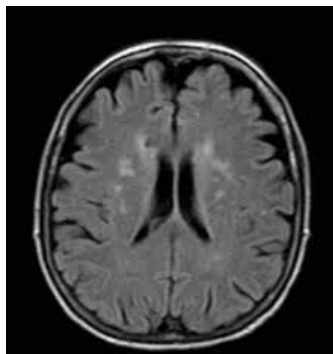
(d) normal brain 4



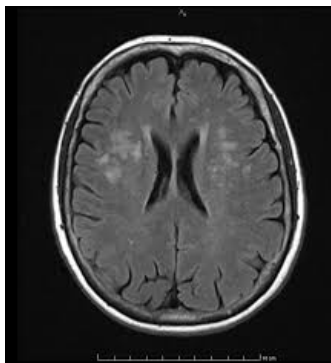
(e) normal brain 5



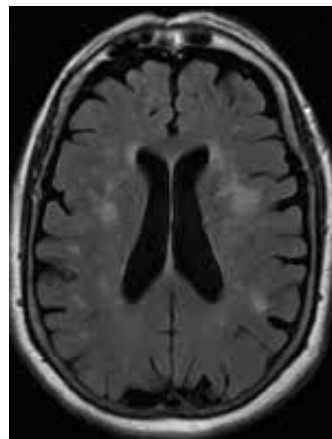
(f) normal brain 6



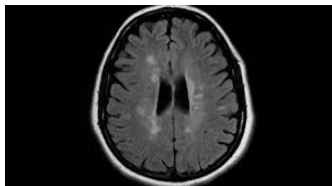
(g) normal brain 7



(h) normal brain 8



(i) normal brain 9



(j) normal brain 10

Figure 4.1: Raw images of normal brain 1-10

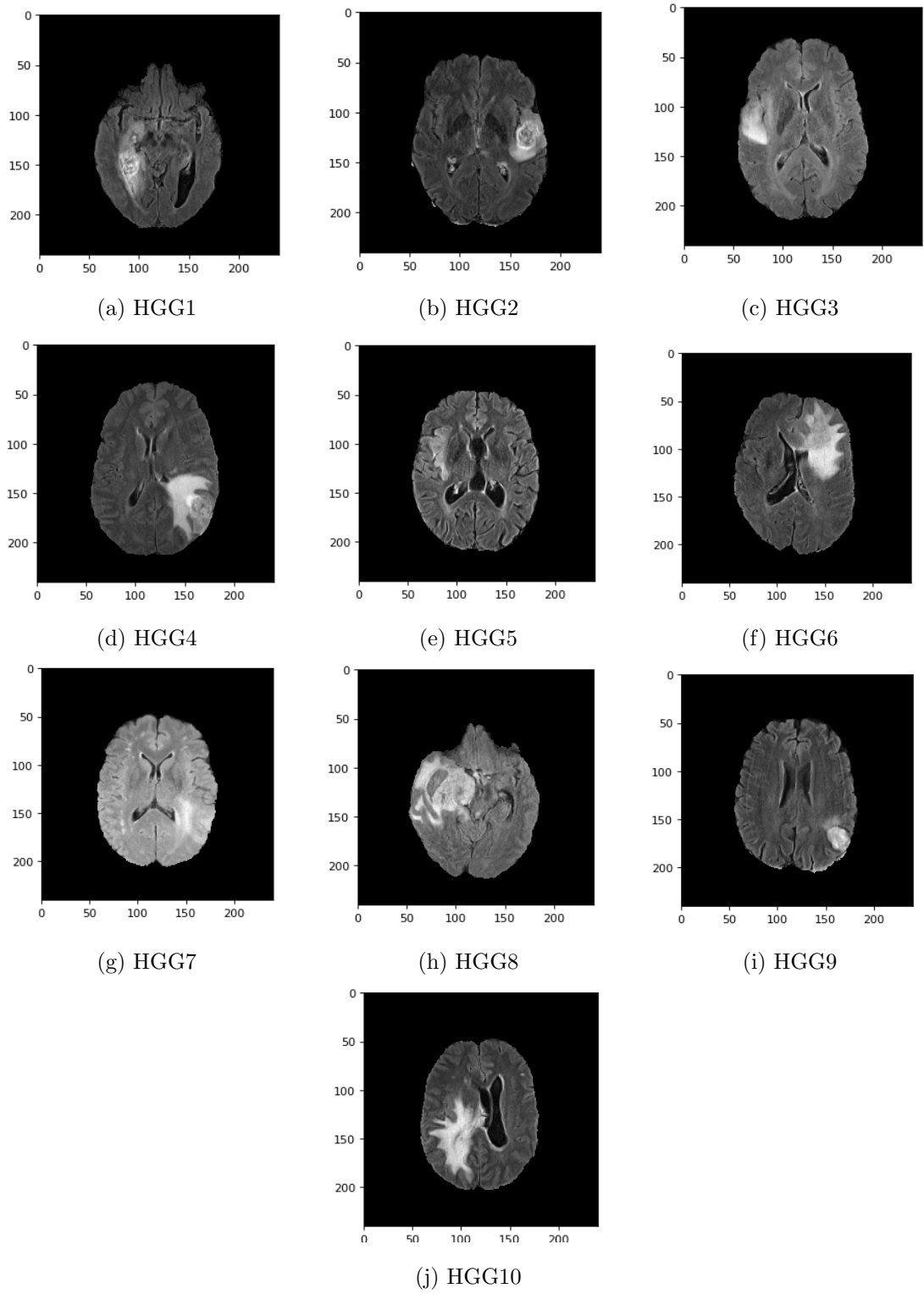


Figure 4.2: Raw images of HGG 1-10

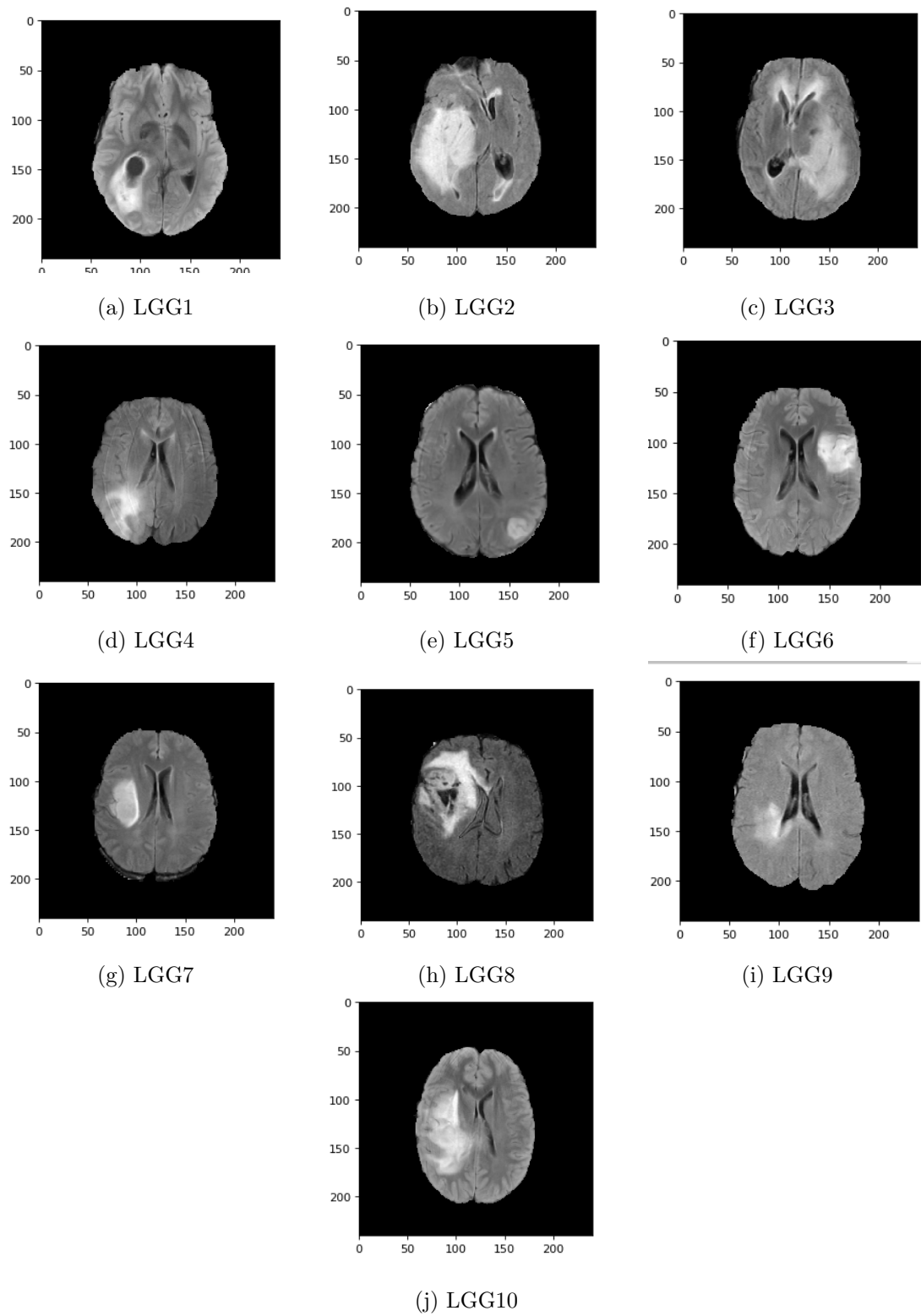


Figure 4.3: Raw images of LGG 1-10

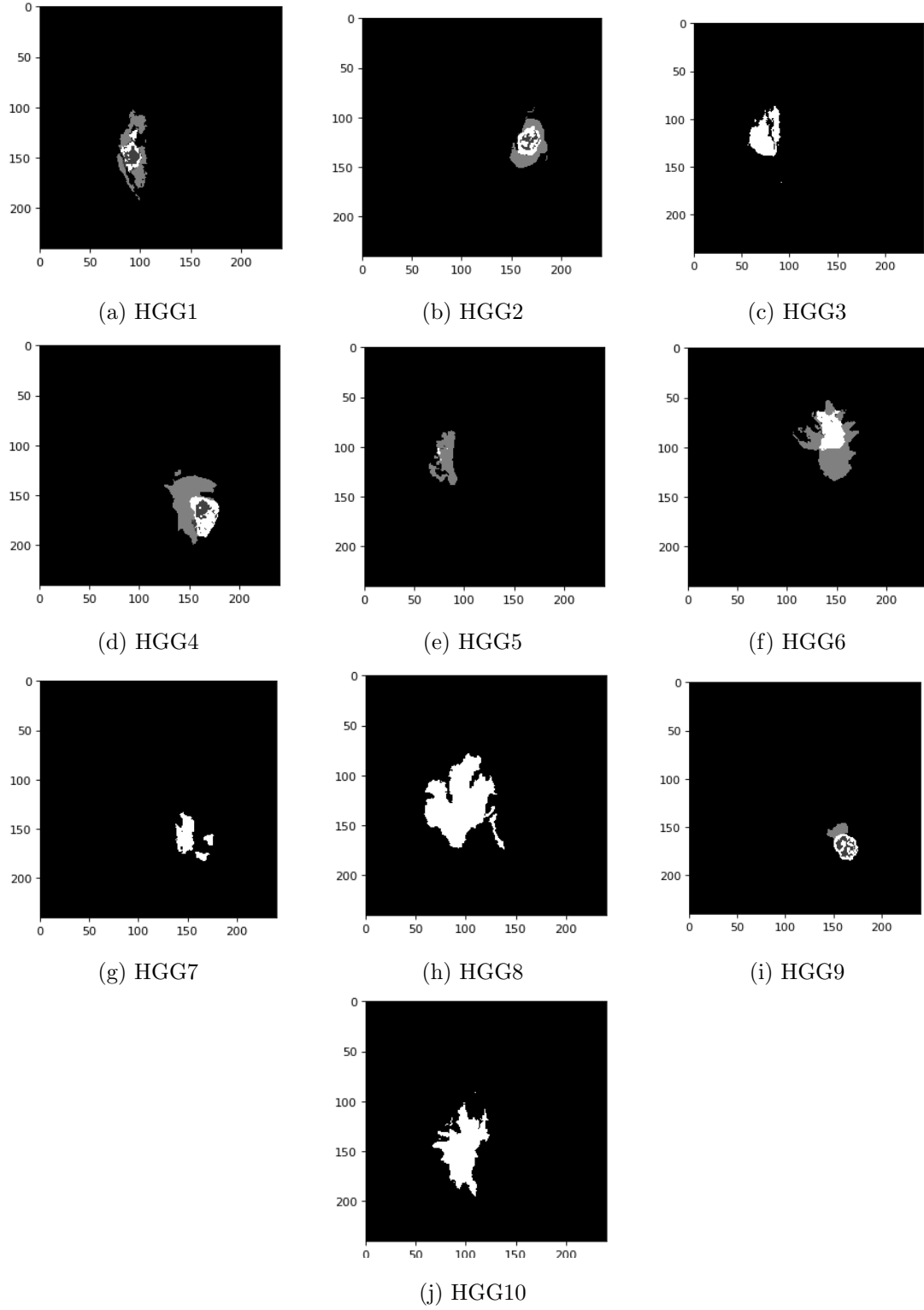


Figure 4.4: Ground truth of HGG 1-10

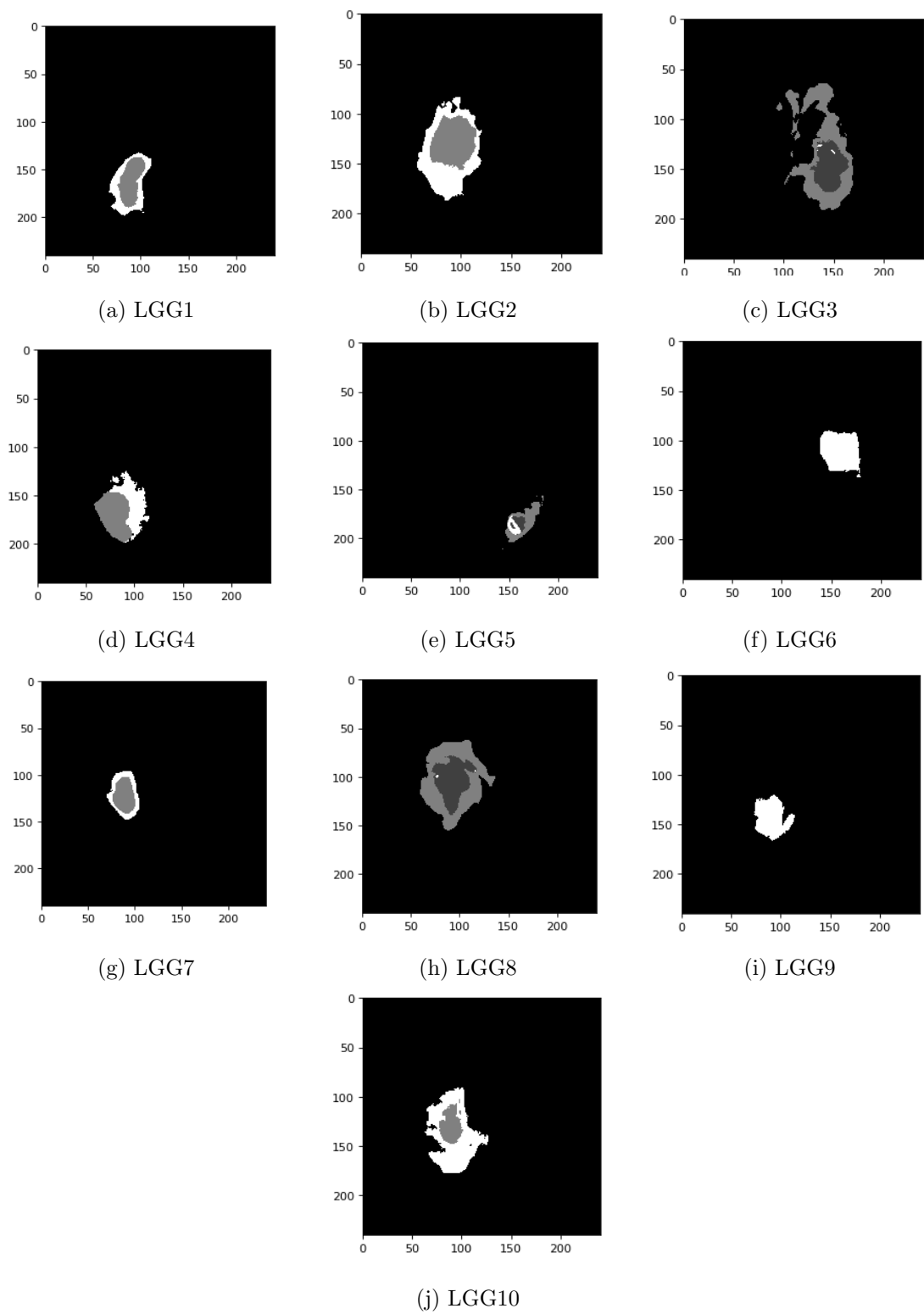
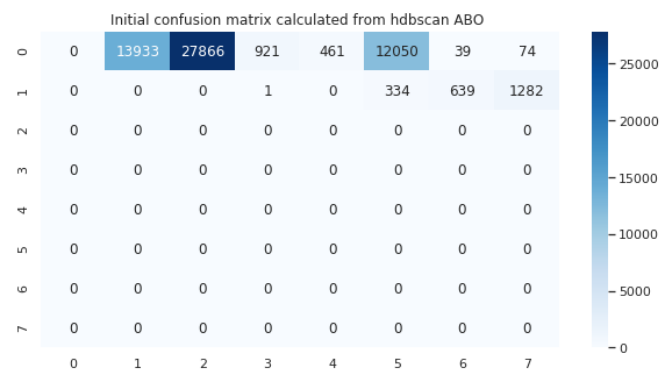


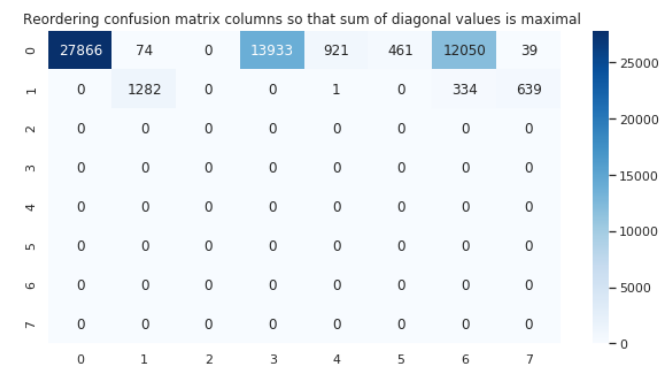
Figure 4.5: Ground truth images of LGG 1-10

i column j contains the number of pixels that are assigned to class i but they have true class j . With a confusion matrix, accuracy can be calculated by dividing the sum of the numbers in diagonal cells by the total number of pixels. The formula of accuracy is between the true class x and the assigned class \hat{x} as shown in Equation 4.1. This equation is commonly used by scikit-learn for accuracy [80]. In the equation, $1(\hat{x}_i = x_i)$ denotes if \hat{x}_i is equal to x_i then, it is equal to one. Otherwise, it is zero. Equation 4.1 needs to be slightly modified since it does not take into account that clustering algorithms do not provide association between class labels and clustered labels[81]. The changes can be seen in equation 4.2. In this equation, $\max_{perm \in P}$ states that P is the set of all permutations in $[1;K]$ where K is the number of clusters. Moreover, $1(perm(\hat{x}_i) = x_i)$ is also changed to take into account the permutations. Figures 4.6 and 4.7 show the initial and reordered confusion matrices for an image. The initial confusion matrix calculates to an accuracy of zero since it needs to be reordered to obtain the maximum value of the sum. Therefore, the reordered matrix in Figures 4.6b and 4.7b show the rearranged columns so that the diagonal values obtained are maximal. For example in Figure 4.6a, 27866, a value in cell (2,0), is shifted to the first column and to the cell (0,0). This can be seen in Figure 4.6b.

Precision measures the fraction of pixels correctly classified as tumor in all the pixels classified as tumor. Equation 4.3 shows the precision formula, in which *true positive* is the number of pixels that are classified as tumor and have true class is tumor, and *false positive* is the number of pixels that are classified as tumor but have non-tumor true class. Recall measures the fraction of pixels correctly classified as tumor in all the pixels whose true class are tumor. Equation 4.4 is the recall formula, in which *false negative* is the number of pixels that are classified as non-tumor but have true class tumor. Precision and recall come at a cost that is they work inversely proportional to each other. One cannot have high precision and high recall at the same time. F1 score was also utilized to show the optimal solution between both precision and recall. Equation 4.5 shows F1 score which combines both precision and recall. While, these are some of the external indices used to show how well the clustering algorithm performs, there were some internal indices used as well to

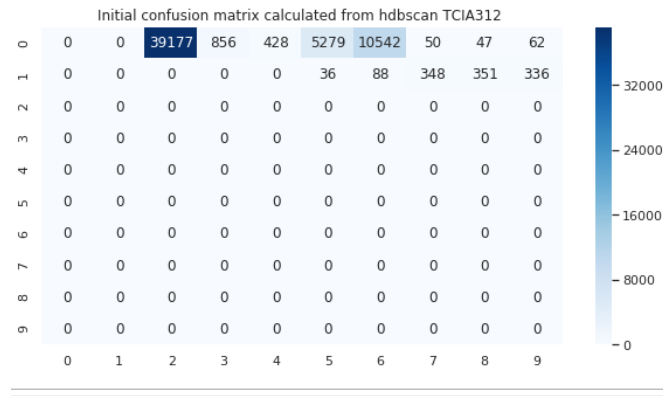


(a)

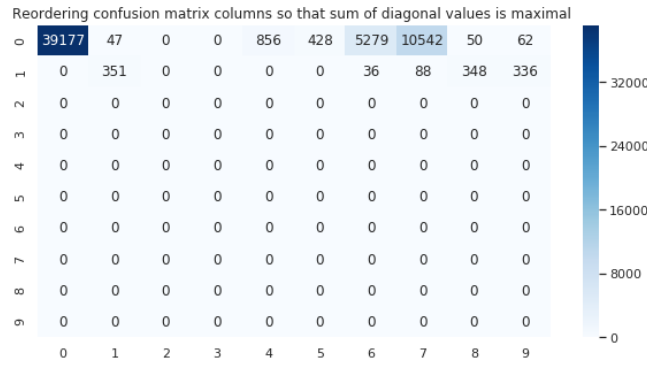


(b)

Figure 4.6: Confusion matrix of a HGG image



(a)



(b)

Figure 4.7: Confusion matrix of a LGG image

determine cluster stability

$$accuracy(x, \hat{x}) = \frac{1}{n} \sum_{i=0}^{n-1} 1(\hat{x}_i = x_i) \quad (4.1)$$

$$accuracy(x, \hat{x}) = \max_{perm \in P} * \frac{1}{n} \sum_{i=0}^{n-1} 1(perm(\hat{x}_i) = x_i) \quad (4.2)$$

$$precision = \frac{true\ positives}{true\ positives + false\ positives} \quad (4.3)$$

$$recall = \frac{true\ positives}{true\ positives + false\ negatives} \quad (4.4)$$

$$F1\ Score = 2 * \frac{precision * recall}{precision + recall} \quad (4.5)$$

4.3 Experimental Results and Analysis

4.3.1 Results of Identifying Tumors

Figures 4.8 and 4.9 show reconstructed images of HGG and LGG clustered by HDB-SCAN. The results reveal the tumor outline clearly. Tumors appear in brighter areas. Figures 4.10 and 4.11 show the results of both HGG and LGG further processed by thresholding, in which tumors are precisely highlighted. Compare the results with the ground truth in Figures 4.4 and Figure 4.5. We can see that in all the 20 images our technique has been able to recognize tumors and differentiate tumors from other tissues.

Our automated technique achieved a precision of 0.88 for HGG and 0.84 for LGG which can be seen in Table 4.1. A higher precision indicates a lower false positive rate. While, we obtained an accuracy of 97.66 using the proposed automated HDB-SCAN+thresholding technique. Other metrics such as recall and F1 score are discussed along with an elaborate comparison with other algorithms in subsection 4.3.3, 4.3.4.

In some images, the tumor areas recognized by our technique have discrepancies with those in the ground truth images. The discrepancies may be due to several reasons. Firstly, HGG tumors are not localized to one area because, malignant tumors are aggressive with rapid and extensive dispersal throughout the brain [82]. An example of this can be seen Figure 4.10e. Another reason they do not look entirely similar is because of MRI artifacts, which is basically an anomaly that should not be present in the image [83]. They can be divided into patient related, signal processing and machine related artifacts. Moreover, it can also be due to leukoaraiosis (white matter hyperintensities), which is common in older patients, as they result in small vessel disease [84]. Small vessel disease in the brain is caused when plaque formation and inflammation affects small cerebral blood vessels. This in turn causes areas to appear white in brains even though they are not tumors. Both artifacts and leukoaraiosis are commonly seen in Figures 4.4, 4.5. Lastly, another anomaly that was seen were frontal horn caps which, are a type of periventricular hyperintensities. They indicate increased water content and reduced tissue (myelin) structure [85]. Thus, are not considered to be pathologically significant. Figure 4.11c displays this as we can see objects around the ventricles that appeared bright after clustering due to this issue.

4.3.2 Results of Processing Normal Images

To examine how our technique can differentiate between normal and brain tumors, the normal brain images were fed into the proposed technique in the same way for tumor images. The output of those images are displayed in Figure 4.12. For all the images, the technique reports no tumor.

In brain MRI images containing ventricles, the ventricle areas have big contrasts with surrounding areas, for example, (d), (e), and (g) in Figure 4.15. To avoid confusion of ventricles with tumors, we use a contour to match ventricles which have higher intensities. An example of such a contour can be seen in Figure 4.13. If an area can be approximately matched by a ventricle contour, it is excluded to be considered as a tumor area.

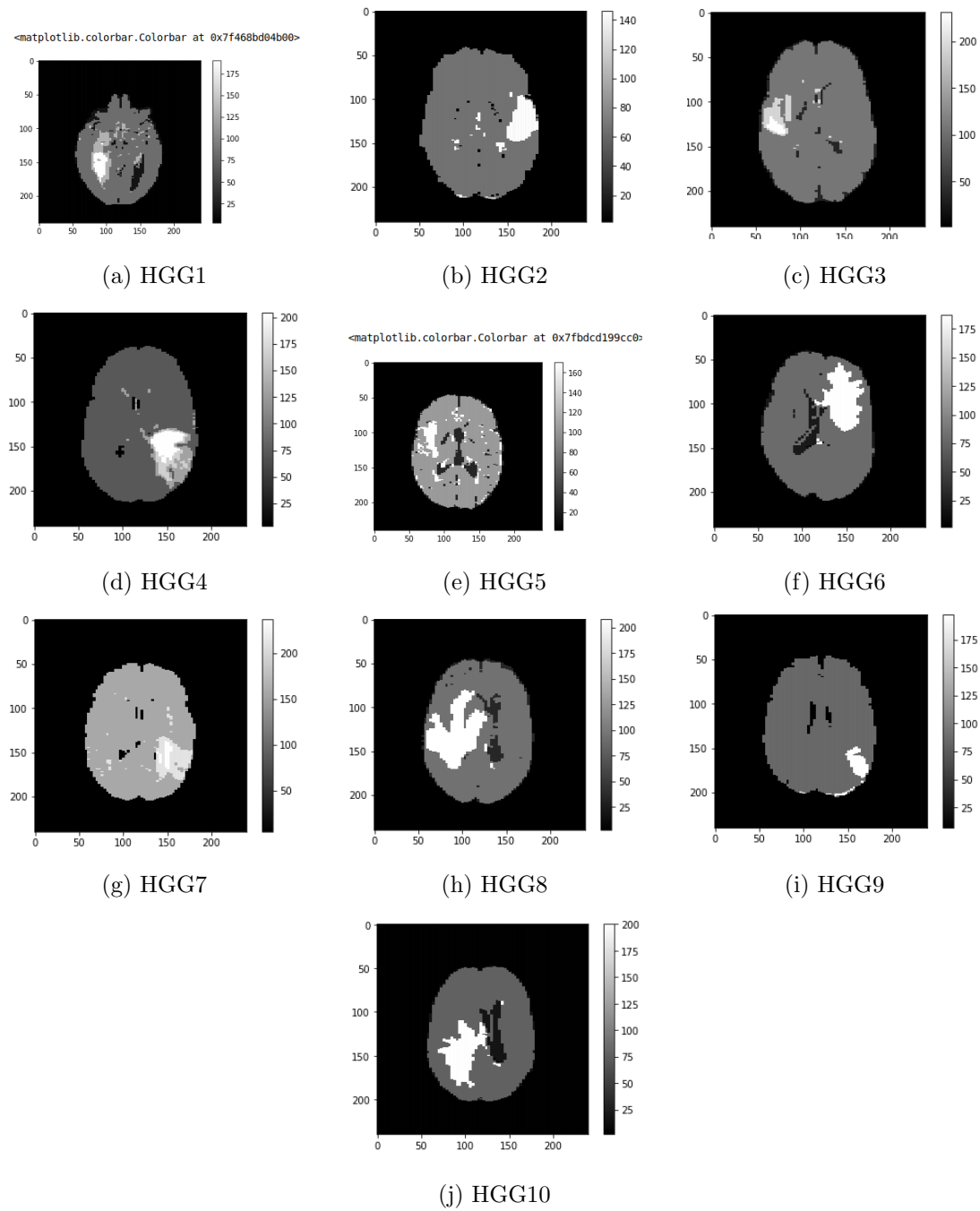


Figure 4.8: Clustered images of HGG 1-10

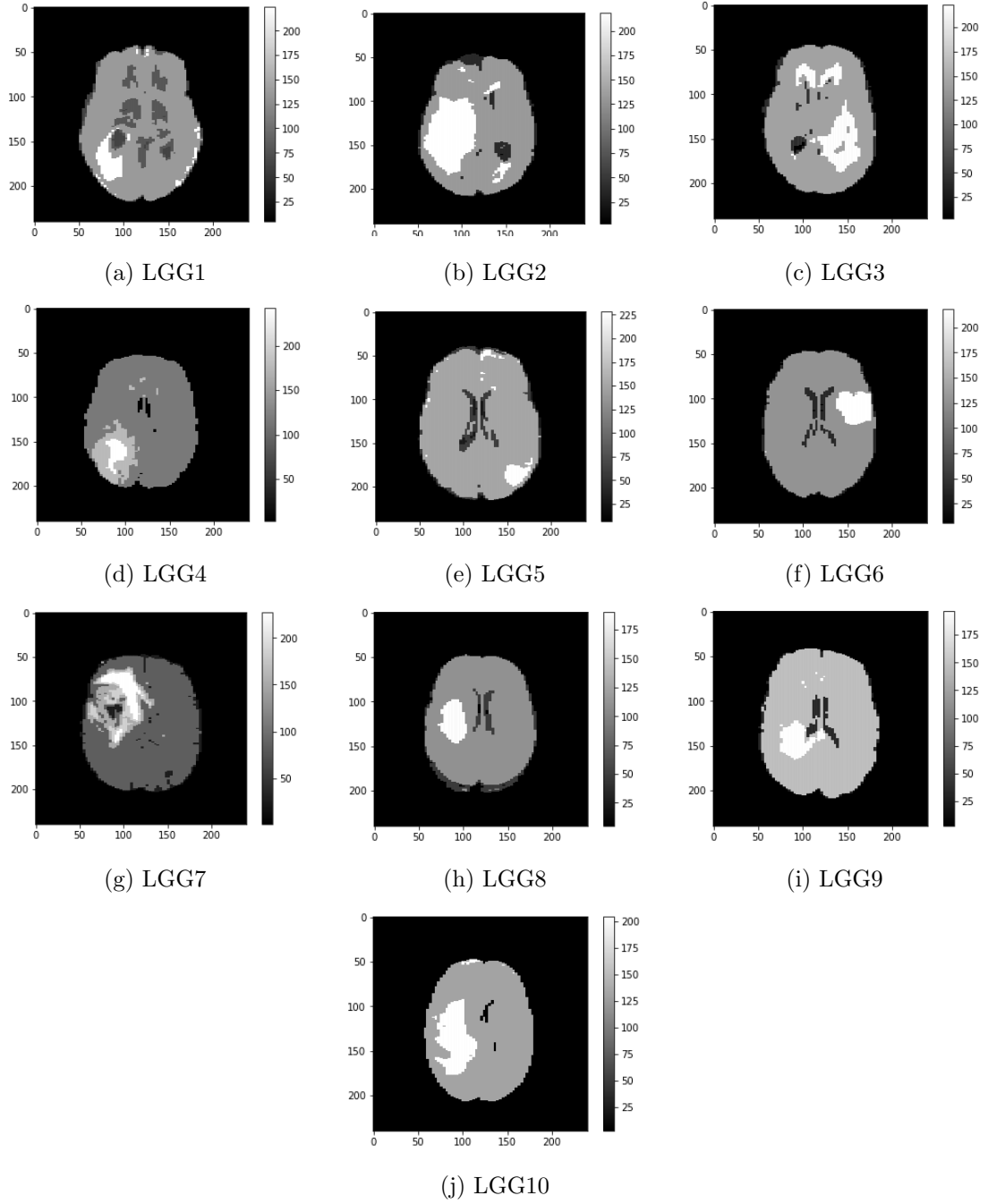


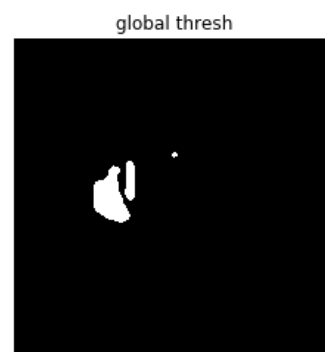
Figure 4.9: Clustered images of LGG 1-10



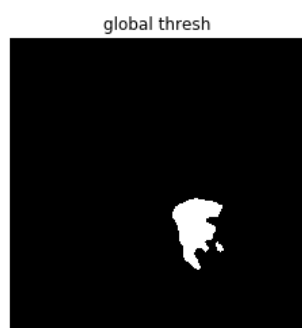
(a) HGG1



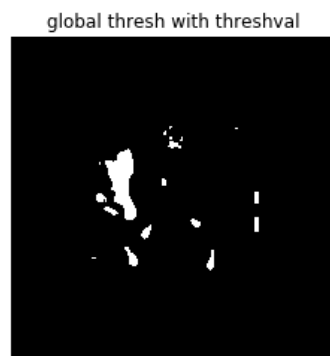
(b) HGG2



(c) HGG3



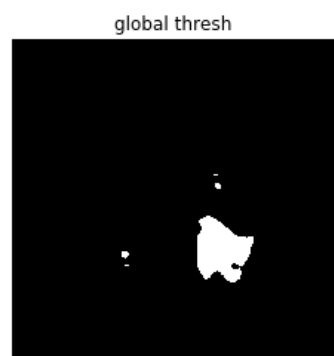
(d) HGG4



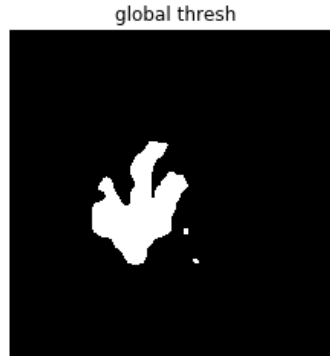
(e) HGG5



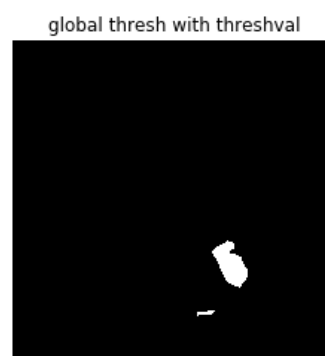
(f) HGG6



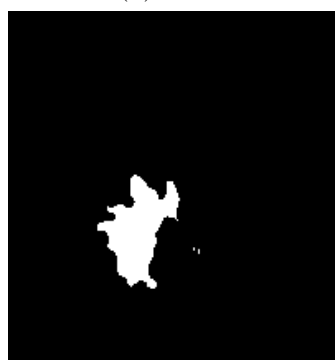
(g) HGG7



(h) HGG8



(i) HGG9



(j) HGG10

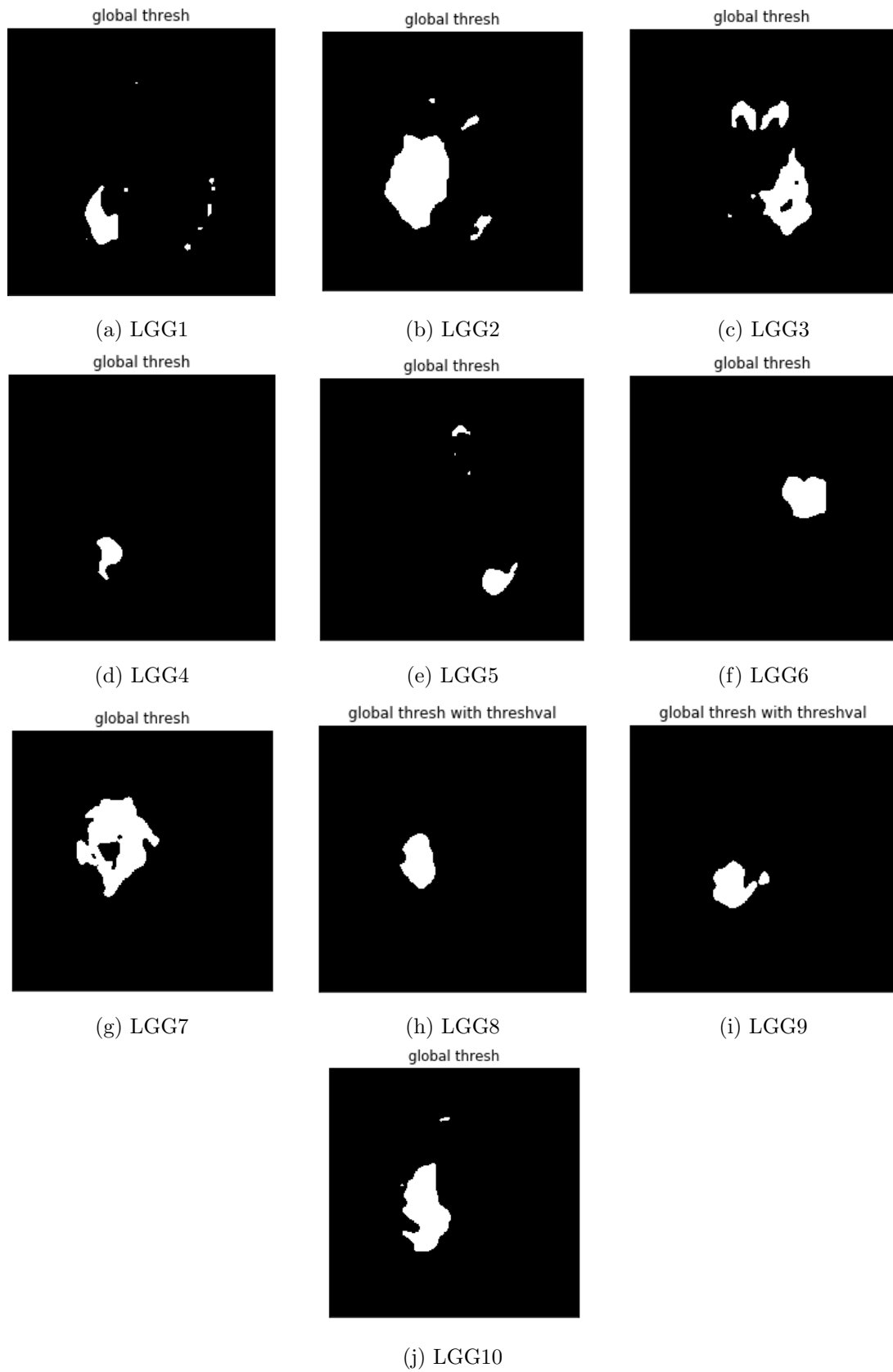


Figure 4.11: Thresholded images of LGG 1-10

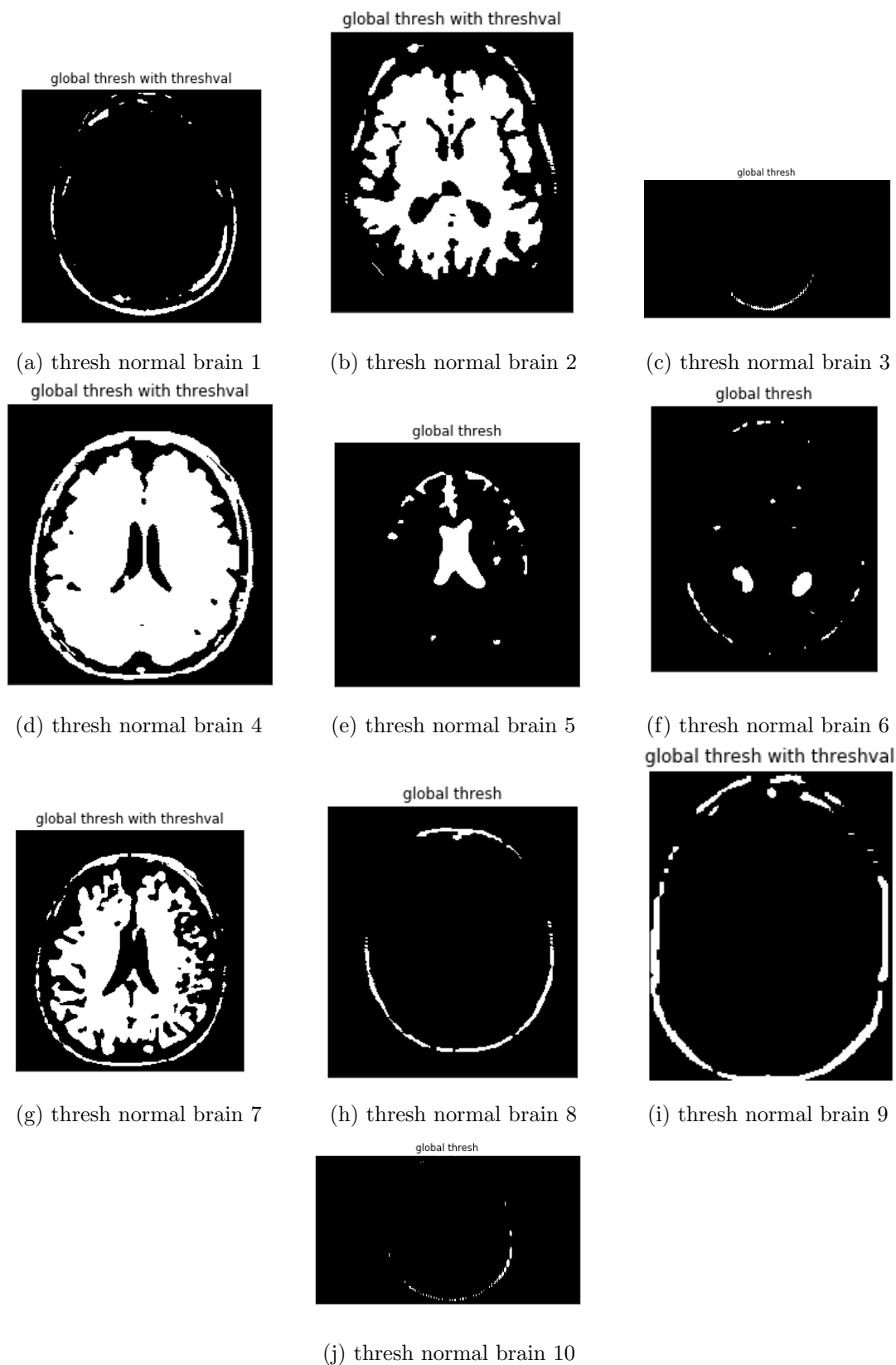


Figure 4.12: Output of normal brain after passing through our system 1-10

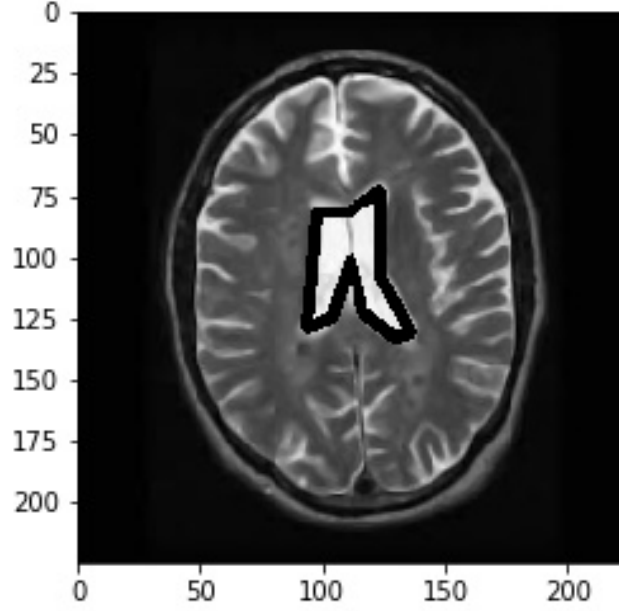


Figure 4.13: Ventricle highlighted

4.3.3 Accuracy Analysis and Comparison

Figure 4.14 shows the accuracy of individual clustering algorithms for comparison. The accuracy of a technique is calculated as an average over the 20 images, using the Equation 4.2. HDBSCAN achieved an accuracy of 73%. K-Means achieved an accuracy of 51.75% , Meanshift got an accuracy of 44%, DBSCAN got an accuracy of 58.5 and Agglomerative got an abysmal accuracy of 30.33%.

The figure shows the poor performance of some clustering algorithms on MRI images. The main reasons for most of them to obtain low accuracies were due to the low intensities and the way the clustering algorithms assemble the clusters. Figure 4.15 illustrates the accuracy and time taken for the total system to run with different clustering algorithms+thresholding on 20 images. The accuracy was calculated using Equation 4.1. K-Means took 13.85s to attain an accuracy of 60%. While, HDBSCAN, took 15.78s and obtained an accuracy of 97.66%. Meanshift took 20s and achieved the accuracy of 51.5%. Meanwhile, DBSCAN took 17.1s to get an accuracy

of 63%. Agglomerative took the longest time to cluster 20 images, 28.99s and achieved a subpar accuracy of 47.5%. We can see that HDBSCAN is the clear winner here for our hybrid model as compared to other clustering algorithms. Although, it is not as fast as compared to K-Means it still obtains a better accuracy than K-Means. Furthermore, the precision, recall and F1 score were calculated using their respective Equations 4.3, 4.4, 4.5. The results are shown in Table 4.1. We obtained optimal precision values which indicates a low false positive rate. While, recall was low due to the inverse relationship between precision and recall. Since our precision is high and recall is significant, our proposed method is able to classify tumors correctly and has the ability to find the relevant tumor region. Finally, a F1 score was calculated to show the average mean between the two metrics.

In addition, a time complexity analysis is presented in Table 4.2 shows how long machine learning methods take to compute an MRI scan [63]. With the advances in U-Net and DF-U-Net, the time cost has decreased to 15.25s. Our proposed method takes only a mere 3s to compute the results of an MRI image. However, machine learning methods have numerous parameters that need to be tuned before classifying the brain MRI image. Our method only takes up to 20 parameters which are much more convenient to tune.

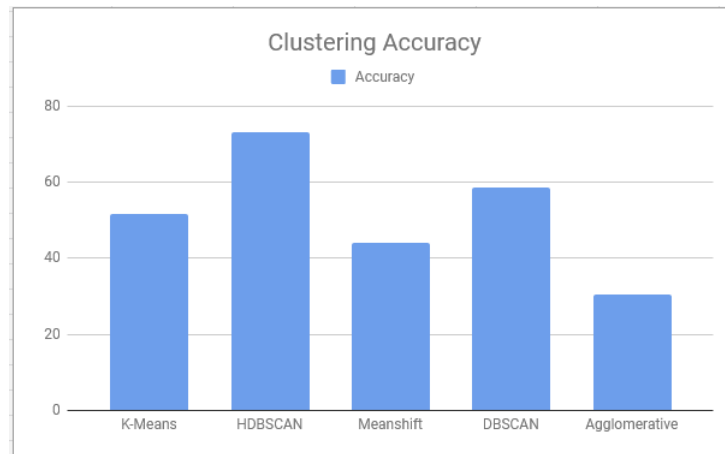


Figure 4.14: Initial clustering accuracy only

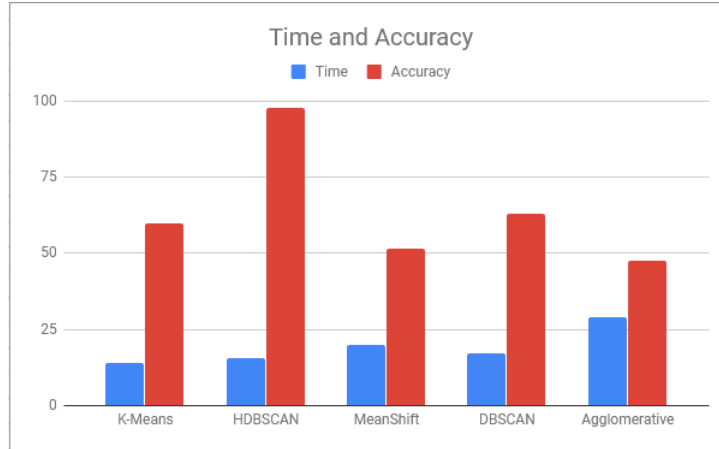


Figure 4.15: Accuracy of the overall system with different algorithms

	Precision	Recall	F1 Score
HGG	0.88	0.80	0.84
LGG	0.84	0.78	0.81

Table 4.1: Precision, Recall and F1 Score of HGG and LGG

Method	Average time cost (per MRI scan)	Number of parameters
DeepMedic	168.84s	84K
MLDeepMedic	186.93s	91K
DF-MLDeepMedic	186.93s	100K
U-Net	15.25s	6.3M
DF-U-Net	15.25s	6.3M
Proposed method	3s	20

Table 4.2: Time complexity of existing machine learning networks

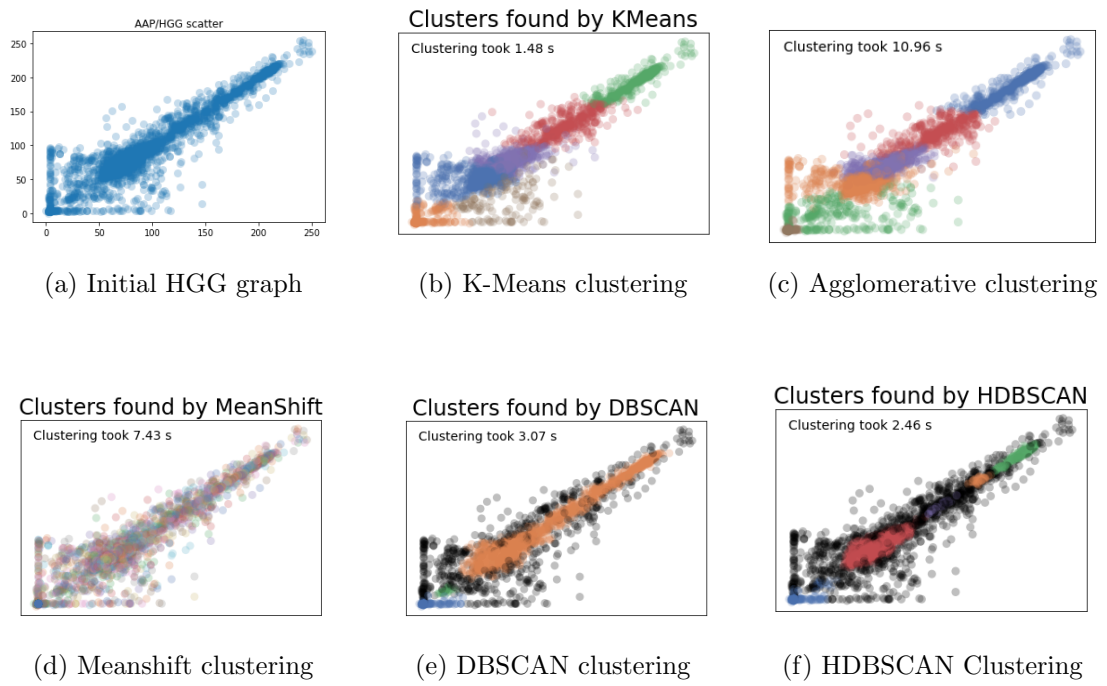


Figure 4.16: Clustering comparison on HGG

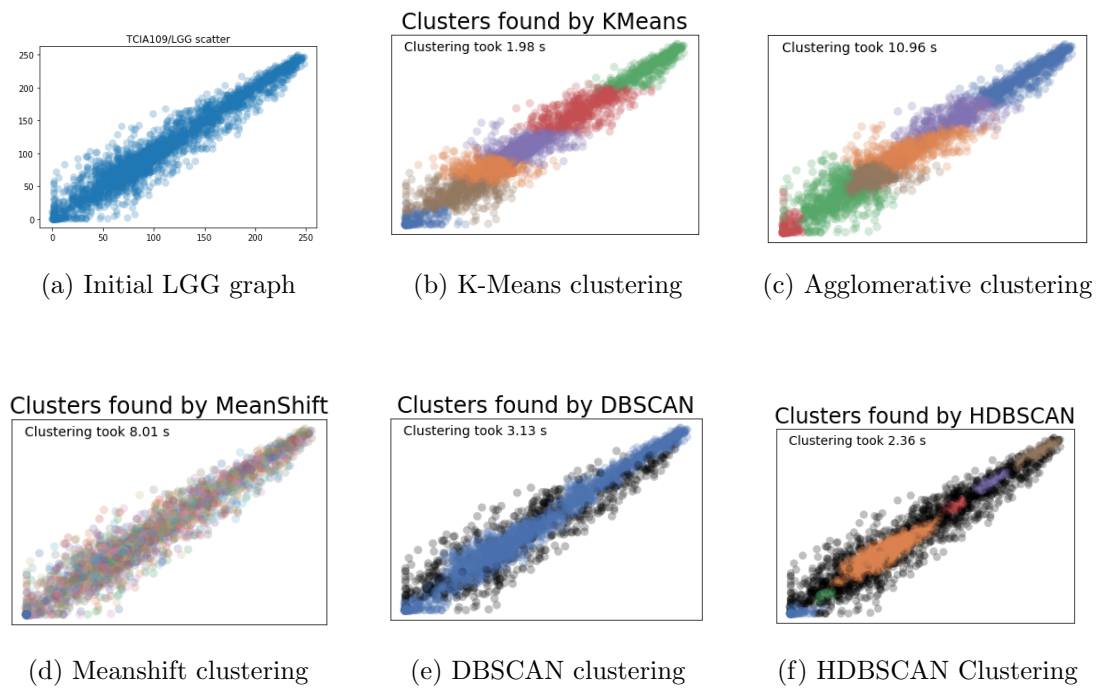


Figure 4.17: Clustering comparing on LGG

4.3.4 Comparison with Other Clustering Algorithms

An initial visual contrast was made to show how clustering algorithms dealt with image points which is demonstrated in Figure 4.16 for HGG and LGG in Figure 4.17. The graphs were created using a transpose on the image array to show that regardless of the position of the points, clustering algorithms worked in the same way. The x and y axis represents the array contents of a (240,240) image. The initial graph and other graphs consist of a 2D array that has been split into two 1D arrays. The initial 1D array contains the rows of the 2D array while, the latter 1D array contains the columns of the 2D array. Since we cannot pass two points in a scatter plot, the x is set as a (240,) array while, y is set as (,240). For example in the initial LGG graph, has the value 140 at (100,70). The points indicate the intensities of the image. The methods can be compared by seeing the division of cluster formation in the different clustering algorithms. This can be clearly seen in any case in both the HGG and LGG graphs. Clusters are represented by different colors and black points indicate noise points. K-Means was the fastest to run but however, included noise points in its clusters. An overlap can be seen especially, in meanshift graphs. Since the points are overlapping each other it is very hard for a flat kernel to effectively segregate the data points into different clusters. Agglomerative took the longest time to run and produced similar results to K-Means. Meanwhile, meanshift could not appropriately cluster points and also clustered noise points. Only DBSCAN and HDBSCAN addressed noise points but, DBSCAN was not able to differentiate points into different clusters according to their intensities. While, HDBSCAN was able to cluster points according to their intensities and took a bit more time to run as compared to K-Means.

Furthermore, an elaborate comparison was made against other state of the art techniques is illustrated in Table 4.3. Papageorgiou et al. classified brain tumor into high and low grade and achieved an overall accuracy of 92% [68]. Zachraki et al. achieved an accuracy of 80% and used only 102 images of glioblastomas [86]. Kharrat et al. attained an accuracy of 97% from binary classifying 83 images [87]. Moreover, Sachdeva et al. improved upon the accuracy from 91.7 to 94% by using augmented

Method / Year	Dataset(# of Images)	Accuracy(%)
Papageorgiou et al. [68] / 2008	100	92
Zachraki et al. [86] / 2009	102	80
Kharrat et al. [87] / 2010	186	94.44
Vu et al. [91] / 2015	190	94.64
Sachdeva et al. [88] / 2016	428	94
Pinaya et al. [89] / 2016	231	73.6
Barker et al. [90] / 2016	302	93.1
Hsieh et al. [92] / 2017	109	88
Sajjad et al. [66] / 2019	3630	90.67
Amin kabir et al. [67] / 2019	329	96.5
Proposed method	351	97.66

Table 4.3: Comparison against recent papers

images [88]. Pinaya et al. proposed a traditional support vector machine that was implemented along side a deep belief network [89]. However, the method attained an accuracy of 73.6% . In another method, Barker et al. classified several pathological images into LGG and HGG achieving an accuracy of 93% [90]. While, Vu et al. classified histopathological images with and without micro vascular proliferation [91]. The method acquired an accuracy of 94.64%. Likewise, Hsieh et al. four global histograms along with local texture features for the classification of LGG and HGG [92]. Sajjad et al. proposed a multi-grade brain tumor classification which was made using a VGG-19 [66]. They did several augmentations on their dataset and performed analysis before and after augmentation. They attained an accuracy of 87.38% without augmentation and accuracy increased to 90.67% after augmentation. While, Amin kabir et al. implemented a CNN along with a genetic algorithm approach to evolve the hyperparameters based on the generations [67]. Lastly, our method attained an accuracy of 97.66 which combines HDBSCAN and thresholding together.

This chapter shows how our automated hybrid system works and can effectively

identify tumors without human interaction. Moreover, differentiation between normal brains and tumor images was also demonstrated to show how the system deals with normal brain images. We were able to achieve an accuracy of 97.66% on both HGG and LGG images with our automated method.

Chapter 5

Summary and Future Work

5.1 Summary of the Research

This research contributed a new technique for recognizing brain tumors in MRI images. By integrating the clustering algorithms of HDBSCAN and thresholding, this technique has advantages of both supervised and unsupervised classification/clustering: It may recognize tumors without involvement of a human operator. Experimental results were promising: Brain tumor recognition was automated, while accuracy and efficiency were competitive.

Brain cancer is a prevalent disease that affects Canadians regardless of age. Identification of brain tumors from MRI images is often difficult. Noises, bias fields, and overlapping of tumor and non-tumor regions make the task even harder. Our contribution can serve as a starting point for developing automatic brain tumor recognition systems, which may help doctors diagnose tumors more effectively and efficiently, and in turn may allow more patients to be treated in time.

5.2 Future Work

This research can be improved in many ways. We may add a layer or stage after recognizing a tumor, to determine if the lesion is benign or malignant. This can be done by examining the shape of the tumor if it is irregular, the amount of necrosis, the location of tumor and vascular supply [93][94]. This is achievable by incorporating more advanced techniques of pattern recognition and machine learning.

Currently, a problem with classifying LGG and HGG is that the guidelines keep changing in terms of how the tumor can be identified [95]. The World Health Organi-

zation (WHO) recently updated their guidelines where they restructured how gliomas are supposed to be classified and added new mutation markers. They created a four layer approach before arriving at the final diagnosis for the patient. Since these are significant changes in guidelines, the models will have to be re-evaluated to see if it fits the standards of radiologists.

Furthermore, molecular evaluation was also included which may tell us with a greater confidence if a tumor is malignant or benign. The knowledge of molecular evaluation could also be added into the tumor recognition to improve accuracy.

Bibliography

- [1] L. Smith, R. Woods, D. Brenner, S. Bryan, C. Louzada, A. Shaw, D. Turner, H. Weir K., A. Demers, L. Ellison, and M. Dixon. Canadian cancer statistics 2019, 2019.
- [2] L. McInnes, J. Healy, and S. Astels. Hdbscan documentation, Dec 2019.
- [3] D. Persaud-Sharma, J. Burns, J. Trangle, and S. Moulik. Disparities in brain cancer in the united states: a literature review of gliomas. *Medical Sciences*, 5(3):16, 2017.
- [4] A. P. Brady. Error and discrepancy in radiology: inevitable or avoidable? *Insights into imaging*, 8(1):171–182, 2017.
- [5] S. Chaplot, L. Patnaik, and N. Jagannathan. Classification of magnetic resonance brain images using wavelets as input to support vector machine and neural network. *Biomedical signal processing and control*, 1(1):86–92, 2006.
- [6] D. C. Preston. Mri basics, 2006.
- [7] S. M. Smith. Fast robust automated brain extraction. *Human brain mapping*, 17(3):143–155, 2002.
- [8] M. Battaglini, S. M. Smith, S. Brogi, and N. De Stefano. Enhanced brain extraction improves the accuracy of brain atrophy estimation. *Neuroimage*, 40(2):583–589, 2008.
- [9] F. Ségonne, A. M. Dale, E. Busa, M. Glessner, D. Salat, H. K. Hahn, and B. Fischl. A hybrid approach to the skull stripping problem in mri. *Neuroimage*, 22(3):1060–1075, 2004.
- [10] J. G. Sled, A. P. Zijdenbos, and A. C. Evans. A nonparametric method for automatic correction of intensity nonuniformity in mri data. *IEEE transactions on medical imaging*, 17(1):87–97, 1998.
- [11] C. M. Collins, W. Liu, W. Schreiber, Q. X. Yang, and M. B. Smith. Central brightening due to constructive interference with, without, and despite dielectric

- resonance. *Journal of Magnetic Resonance Imaging: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 21(2):192–196, 2005.
- [12] J.-F. Mangin. Entropy minimization for automatic correction of intensity nonuniformity. In *Proceedings IEEE Workshop on Mathematical Methods in Biomedical Image Analysis. MMBIA-2000 (Cat. No. PR00737)*, pages 162–169. IEEE, 2000.
 - [13] E. B. Lewis and N. C. Fox. Correction of differential intensity inhomogeneity in longitudinal mr images. *Neuroimage*, 23(1):75–83, 2004.
 - [14] 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*, 58(13):R97, 2013.
 - [15] N. Otsu. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1):62–66, 1979.
 - [16] G. W. Zack, W. E. Rogers, and S. Latt. Automatic measurement of sister chromatid exchange frequency. *Journal of Histochemistry & Cytochemistry*, 25(7):741–753, 1977.
 - [17] L.-K. Huang and M.-J. J. Wang. Image thresholding by minimizing the measures of fuzziness. *Pattern recognition*, 28(1):41–51, 1995.
 - [18] J. H. Ward Jr. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244, 1963.
 - [19] B. Szmrecsanyi. *Grammatical variation in British English dialects: A study in corpus-based dialectometry*. Cambridge University Press, 2012.
 - [20] J. C. Dunn. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. 1973.
 - [21] J. C. Bezdek. Objective function clustering. In *Pattern recognition with fuzzy objective function algorithms*, pages 43–93. Springer, 1981.
 - [22] M. De Craene, A. du Bois d’Aische, B. Macq, and S. K. Warfield. Multi-subject registration for unbiased statistical atlas construction. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 655–662. Springer, 2004.

- [23] C. J. Twining, T. Cootes, S. Marsland, V. Petrovic, R. Schestowitz, and C. J. Taylor. A unified information-theoretic approach to groupwise non-rigid registration and model building. In *Biennial International Conference on Information Processing in Medical Imaging*, pages 1–14. Springer, 2005.
- [24] M. B. Cuadra, V. Duay, and J.-P. Thiran. Atlas-based segmentation. In *Handbook of Biomedical Imaging*, pages 221–244. Springer, 2015.
- [25] J. C. Gee, M. Reivich, and R. Bajcsy. Elastically deforming a three-dimensional atlas to match anatomical brain images. 1993.
- [26] M. R. Sabuncu, B. T. Yeo, K. Van Leemput, B. Fischl, and P. Golland. A generative model for image segmentation based on label fusion. *IEEE transactions on medical imaging*, 29(10):1714–1729, 2010.
- [27] S. Bauer, C. Seiler, T. Bardyn, P. Buechler, and M. Reyes. Atlas-based segmentation of brain tumor images using a markov random field-based tumor growth model and non-rigid registration. In *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology*, pages 4080–4083. IEEE, 2010.
- [28] Y. Zhou and J. Bai. Atlas-based fuzzy connectedness segmentation and intensity nonuniformity correction applied to brain mri. *IEEE transactions on biomedical engineering*, 54(1):122–129, 2006.
- [29] S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations. *Journal of computational physics*, 79(1):12–49, 1988.
- [30] S. Osher and R. P. Fedkiw. Level set methods: an overview and some recent results. *Journal of Computational physics*, 169(2):463–502, 2001.
- [31] S. Osher and N. Paragios. *Geometric level set methods in imaging, vision, and graphics*. Springer Science & Business Media, 2003.
- [32] K. Thapaliya, J.-Y. Pyun, C.-S. Park, and G.-R. Kwon. Level set method with automatic selective local statistics for brain tumor segmentation in mr images. *Computerized Medical Imaging and Graphics*, 37(7-8):522–537, 2013.
- [33] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International journal of computer vision*, 1(4):321–331, 1988.

- [34] D. Terzopoulos, A. Witkin, and M. Kass. Constraints on deformable models: Recovering 3d shape and nonrigid motion. *Artificial intelligence*, 36(1):91–123, 1988.
- [35] L. D. Cohen and I. Cohen. Finite-element methods for active contour models and balloons for 2-d and 3-d images. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (11):1131–1147, 1993.
- [36] E. Ilunga-Mbuyamba, J. G. Avina-Cervantes, A. Garcia-Perez, R. de Jesus Romero-Troncoso, H. Aguirre-Ramos, I. Cruz-Aceves, and C. Chalopin. Localized active contour model with background intensity compensation applied on automatic mr brain tumor segmentation. *Neurocomputing*, 220:84–97, 2017.
- [37] C. Vijayakumar and D. C. Gharpure. Development of image-processing software for automatic segmentation of brain tumors in mr images. *Journal of medical physics/Association of Medical Physicists of India*, 36(3):147, 2011.
- [38] J. M. Lötjönen, R. Wolz, J. R. Koikkalainen, L. Thurfjell, G. Waldemar, H. Soininen, D. Rueckert, A. D. N. Initiative, et al. Fast and robust multi-atlas segmentation of brain magnetic resonance images. *Neuroimage*, 49(3):2352–2365, 2010.
- [39] A. R. Raju, S. Pabboju, and R. R. Rao. Hybrid active contour model and deep belief network based approach for brain tumor segmentation and classification. *Sensor Review*, 2019.
- [40] A. Ortiz, J. Gorriz, J. Ramirez, and D. Salas-Gonzalez. Improving mr brain image segmentation using self-organising maps and entropy-gradient clustering. *Information Sciences*, 262:117–136, 2014.
- [41] A. Akselrod-Ballin, M. Galun, M. J. Gomori, R. Basri, and A. Brandt. Atlas guided identification of brain structures by combining 3d segmentation and svm classification. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 209–216. Springer, 2006.
- [42] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.
- [43] D. Ciresan, A. Giusti, L. M. Gambardella, and J. Schmidhuber. Deep neural networks segment neuronal membranes in electron microscopy images. In *Advances in neural information processing systems*, pages 2843–2851, 2012.

- [44] G. Urban, M. Bendszus, F. Hamprecht, and J. Kleesiek. Multi-modal brain tumor segmentation using deep convolutional neural networks. *MICCAI BraTS (Brain Tumor Segmentation) Challenge. Proceedings, winning contribution*, pages 31–35, 2014.
- [45] M. Havaei, A. Davy, D. Warde-Farley, A. Biard, A. Courville, Y. Bengio, C. Pal, P.-M. Jodoin, and H. Larochelle. Brain tumor segmentation with deep neural networks. *Medical image analysis*, 35:18–31, 2017.
- [46] S. Pereira, A. Pinto, V. Alves, and C. A. Silva. Brain tumor segmentation using convolutional neural networks in mri images. *IEEE transactions on medical imaging*, 35(5):1240–1251, 2016.
- [47] U. M. Braga-Neto and E. R. Dougherty. Is cross-validation valid for small-sample microarray classification? *Bioinformatics*, 20(3):374–380, 2004.
- [48] A. Isaksson, M. Wallman, H. Göransson, and M. G. Gustafsson. Cross-validation and bootstrapping are unreliable in small sample classification. *Pattern Recognition Letters*, 29(14):1960–1965, 2008.
- [49] K. M. Iftekharuddin, J. Zheng, M. A. Islam, and R. J. Ogg. Fractal-based brain tumor detection in multimodal mri. *Applied Mathematics and Computation*, 207(1):23–41, 2009.
- [50] J. Yao. Image processing in tumor imaging. *New techniques in oncologic imaging*, pages 79–102, 2006.
- [51] S. D. Olabarriaga and A. W. Smeulders. Interaction in the segmentation of medical images: A survey. *Medical image analysis*, 5(2):127–142, 2001.
- [52] D. R. White, A. S. Houston, W. F. Sampson, and G. P. Wilkins. Intra-and interoperator variations in region-of-interest drawing and their effect on the measurement of glomerular filtration rates. *Clinical nuclear medicine*, 24(3):177–181, 1999.
- [53] M. Visser, D. Müller, R. van Duijn, M. Smits, N. Verburg, E. Hendriks, R. Nabuurs, J. Bot, R. Eijgelaar, M. Witte, et al. Inter-rater agreement in glioma segmentations on longitudinal mri. *NeuroImage: Clinical*, 22:101727, 2019.
- [54] S. Luo, R. Li, and S. Ourselin. A new deformable model using dynamic gradient vector flow and adaptive balloon forces. In *APRS Workshop on Digital Image Computing, Brisbane, Australia*. Citeseer, 2003.

- [55] S. Bauer, L.-P. Nolte, and M. Reyes. Fully automatic segmentation of brain tumor images using support vector machine classification in combination with hierarchical conditional random field regularization. In *international conference on medical image computing and computer-assisted intervention*, pages 354–361. Springer, 2011.
- [56] A. Pinto, S. Pereira, D. Rasteiro, and C. A. Silva. Hierarchical brain tumour segmentation using extremely randomized trees. *Pattern Recognition*, 82:105–117, 2018.
- [57] D. Zikic, B. Glocker, E. Konukoglu, A. Criminisi, C. Demiralp, J. Shotton, O. M. Thomas, T. Das, R. Jena, and S. J. Price. Decision forests for tissue-specific segmentation of high-grade gliomas in multi-channel mr. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 369–376. Springer, 2012.
- [58] B. Caldaïrou, N. Passat, P. A. Habas, C. Studholme, and F. Rousseau. A non-local fuzzy segmentation method: application to brain mri. *Pattern Recognition*, 44(9):1916–1927, 2011.
- [59] A. Gharipour and A. W.-C. Liew. Segmentation of cell nuclei in fluorescence microscopy images: An integrated framework using level set segmentation and touching-cell splitting. *Pattern Recognition*, 58:1–11, 2016.
- [60] Q. Zhu, B. Du, P. Yan, H. Lu, and L. Zhang. Shape prior constrained pso model for bladder wall mri segmentation. *Neurocomputing*, 294:19–28, 2018.
- [61] M. Prastawa, E. Bullitt, N. Moon, K. Van Leemput, and G. Gerig. Automatic brain tumor segmentation by subject specific modification of atlas priors1. *Academic radiology*, 10(12):1341–1348, 2003.
- [62] A. Işın, C. Direkoğlu, and M. Şah. Review of mri-based brain tumor image segmentation using deep learning methods. *Procedia Computer Science*, 102:317–324, 2016.
- [63] S. Chen, C. Ding, and M. Liu. Dual-force convolutional neural networks for accurate brain tumor segmentation. *Pattern Recognition*, 88:90–100, 2019.
- [64] E. A. AlBadawy, A. Saha, and M. A. Mazurowski. Deep learning for segmentation of brain tumors: Impact of cross-institutional training and testing. *Medical physics*, 45(3):1150–1158, 2018.

- [65] N. Abiwinanda, M. Hanif, S. T. Hesaputra, A. Handayani, and T. R. Mengko. Brain tumor classification using convolutional neural network. In *World Congress on Medical Physics and Biomedical Engineering 2018*, pages 183–189. Springer, 2019.
- [66] M. Sajjad, S. Khan, K. Muhammad, W. Wu, A. Ullah, and S. W. Baik. Multi-grade brain tumor classification using deep cnn with extensive data augmentation. *Journal of computational science*, 30:174–182, 2019.
- [67] A. K. Anaraki, M. Ayati, and F. Kazemi. Magnetic resonance imaging-based brain tumor grades classification and grading via convolutional neural networks and genetic algorithms. *Biocybernetics and Biomedical Engineering*, 39(1):63–74, 2019.
- [68] E. Papageorgiou, P. Spyridonos, D. T. Glotsos, C. D. Stylios, P. Ravazoula, G. Nikiforidis, and P. P. Groumpos. Brain tumor characterization using the soft computing technique of fuzzy cognitive maps. *Applied Soft Computing*, 8(1):820–828, 2008.
- [69] B. Menze, A. Jakab, S. Bauer, J. Kalpathy-Cramer, K. Farahani, and J. Kirby. The multimodal brain tumor image segmentation benchmark (brats)”. *IEEE Transactions on Medical Imaging*, 34(10):19932024.
- [70] S. Bakas, H. Akbari, A. Sotiras, M. Bilello, M. Rozycki, and J. Kirby. Advancing the cancer genome atlas glioma mri collections with expert segmentation labels and radiomic features”. *Nature Scientific Data*, 4(170117).
- [71] S. Bakas, M. Reyes, A. Jakab, S. Bauer, M. Rempfler, and A. Crimi. Identifying the best machine learning algorithms for brain tumor segmentation. In *Progression Assessment, and Overall Survival Prediction in the BRATS Challenge*”. arXiv preprint arXiv:1811.02629.
- [72] N. Chakrabarty. Brain mri images for brain tumor detection, Apr 2019.
- [73] W. Ma and J.-P. Kruth. Parameterization of randomly measured points for least squares fitting of b-spline curves and surfaces. *Computer-Aided Design*, 27(9):663–675, 1995.
- [74] S. Lee, G. Wolberg, and S. Y. Shin. Scattered data interpolation with multilevel b-splines. *IEEE transactions on visualization and computer graphics*, 3(3):228–244, 1997.

- [75] 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 transactions on medical imaging*, 29(6):1310, 2010.
- [76] Y. Gong and I. F. Sbalzarini. Curvature filters efficiently reduce certain variational energies. *IEEE Transactions on Image Processing*, 26(4):1786–1798, April 2017.
- [77] R. J. Campello, D. Moulavi, and J. Sander. Density-based clustering based on hierarchical density estimates. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 160–172. Springer, 2013.
- [78] S. Bakas, H. Akbari, A. Sotiras, M. Bilello, M. Rozycki, and J. Kirby. Segmentation labels and radiomic features for the pre-operative scans of the tcga-gbm collection”. *The Cancer Imaging Archive*.
- [79] S. Bakas, H. Akbari, A. Sotiras, M. Bilello, M. Rozycki, and J. Kirby. Segmentation labels and radiomic features for the pre-operative scans of the tcga-lgg collection”. *The Cancer Imaging Archive*.
- [80] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [81] F. Role, S. Morbieu, and M. Nadif. Coclust: A python package for co-clustering. 2018.
- [82] R. A. Foty. Tumor cohesion and glioblastoma cell dispersal. *Future oncology*, 9(8):1121–1132, 2013.
- [83] L. Erasmus, D. Hurter, M. Naudé, H. Kritzing, and S. Acho. A short overview of mri artefacts. *SA Journal of Radiology*, 8(2), 2004.
- [84] V. Hachinski, P. Potter, and H. Merskey. Leuko-araiosis: an ancient term for a new problem. *Canadian Journal of Neurological Sciences*, 13(S4):533–534, 1986.
- [85] L. Bronge. Magnetic resonance imaging in dementia: a study of brain white matter changes. *Acta Radiologica*, 43:1–32, 2002.
- [86] E. I. Zacharaki, S. Wang, S. Chawla, D. Soo Yoo, R. Wolf, E. R. Melhem, and C. Davatzikos. Classification of brain tumor type and grade using mri texture

- and shape in a machine learning scheme. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 62(6):1609–1618, 2009.
- [87] A. Kharrat, K. Gasmi, M. B. Messaoud, N. Benamrane, and M. Abid. A hybrid approach for automatic classification of brain mri using genetic algorithm and support vector machine. *Leonardo journal of sciences*, 17(1):71–82, 2010.
 - [88] J. Sachdeva, V. Kumar, I. Gupta, N. Khandelwal, and C. K. Ahuja. A package-sfercb-segmentation, feature extraction, reduction and classification analysis by both svm and ann for brain tumors. *Applied soft computing*, 47:151–167, 2016.
 - [89] W. H. Pinaya, A. Gadelha, O. M. Doyle, C. Noto, A. Zugman, Q. Cordeiro, A. P. Jackowski, R. A. Bressan, and J. R. Sato. Using deep belief network modelling to characterize differences in brain morphometry in schizophrenia. *Scientific reports*, 6:38897, 2016.
 - [90] J. Barker, A. Hoogi, A. Depeursinge, and D. L. Rubin. Automated classification of brain tumor type in whole-slide digital pathology images using local representative tiles. *Medical image analysis*, 30:60–71, 2016.
 - [91] T. H. Vu, H. S. Mousavi, V. Monga, G. Rao, and U. A. Rao. Histopathological image classification using discriminative feature-oriented dictionary learning. *IEEE transactions on medical imaging*, 35(3):738–751, 2015.
 - [92] K. L.-C. Hsieh, C.-M. Lo, and C.-J. Hsiao. Computer-aided grading of gliomas based on local and global mri features. *Computer methods and programs in biomedicine*, 139:31–38, 2017.
 - [93] K. A. Rajasekaran and C. C. Gounder. Advanced brain tumour segmentation from mri images. *High-Resolution Neuroimaging: Basic Physical Principles and Clinical Applications*, page 83, 2018.
 - [94] D. Johnson, F. Diehn, C. Giannini, R. B. Jenkins, S. Jenkins, I. F. Parney, and T. J. Kaufmann. Genetically defined oligodendroglioma is characterized by indistinct tumor borders at mri. *American Journal of Neuroradiology*, 38(4):678–684, 2017.
 - [95] J. Larsen, S. B. Wharton, F. McKeivitt, C. Romanowski, C. Bridgewater, H. Zaki, and N. Hoggard. low grade glioma: an update for radiologists. *The British journal of radiology*, 90(1070):20160600, 2017.

Appendix A

Appendix

$$\mu_{ij} = 1 / \left(\sum_{k=1}^c (d_{ij} / d_{ik}) (2/m - 1) \right) \quad (\text{A.1})$$

$$V_j = \left(\sum_{i=1}^n (\mu_{ij})^m x_i \right) / \left(\sum_{i=1}^n (\mu_{ij})^m \right), \forall_j = 1, 2, \dots, c \quad (\text{A.2})$$

$$d_{mreach-k}(a, b) = \max \{ core_k(a), core_k(b), d(a, b) \} \quad (\text{A.3})$$

$$t_i \leq x_i \leq t_{i+k} \quad i = 1 : n \quad (\text{A.4})$$

	New cases (2019 estimates)			Cases per 100,000		
	Total	Males	Females	Both sexes	Males	Females
All cancers	220,400	113,000	107,400	518.8	559.0	489.5
Lung and bronchus	29,300	14,900	14,500	62.1	66.0	59.6
Breast	27,200	230	26,900	66.8	1.2	128.0
Colorectal	26,300	14,600	11,700	60.6	71.7	50.9
Prostate	22,900	22,900	—	—	118.1	—
Bladder	11,800	9,100	2,700	25.0	42.1	10.6
Non-Hodgkin lymphoma	10,000	5,600	4,400	24.2	29.0	20.0
Thyroid	8,200	2,100	6,100	21.8	11.2	32.1
Melanoma	7,800	4,300	3,500	21.7	25.1	19.1
Kidney and renal pelvis	7,200	4,700	2,500	17.0	23.2	11.3
Uterus (body,NOS)	7,200	—	7,200	—	—	34.5
Leukemia	6,700	4,000	2,700	16.4	20.8	12.5
Pancreas	5,800	3,000	2,800	12.9	14.2	11.7
Oral	5,300	3,700	1,600	12.7	18.4	7.4
Stomach	4,100	2,600	1,450	9.3	13.1	6.1
Multiple myeloma	3,300	1,950	1,400	7.7	9.6	6.0
Brain/CNS	3,000	1,650	1,300	7.1	8.3	6.0
Ovary	3,000	—	3,000	—	—	14.2
Liver	3,000	2,200	780	6.7	10.5	3.2
Esophagus	2,300	1,800	540	5.6	9.2	2.4
Cervix	1,350	—	1,350	—	—	7.2
Larynx	1,150	980	190	2.4	4.2	0.7
Testis	1,150	1,150	—	—	6.4	—
Hodgkin lymphoma	1,000	560	440	2.6	2.9	2.3
All other cancers	21,300	11,000	10,300	47.7	53.4	43.3

Table A.1: Estimated cases for cancers [1]

	Lifetime probability of developing cancer					
	%			One in:		
	Both sexes	Males	Females	Both sexes	Males	Females
All cancers	43.8	44.9	43.0	2.3	2.2	2.3
Lung and bronchus	6.8	7.1	6.6	15	14	15
Colorectal	6.3	7.0	5.6	16	14	18
Breast	6.2	0.1	12.1	16	880	8
Prostate	—	11.3	—	—	9	—
Bladder	2.9	4.5	1.3	35	22	75
Non-Hodgkin lymphoma	2.4	2.7	2.2	41	37	46
Melanoma	2.1	2.4	1.8	48	42	56
Leukemia	1.7	2.0	1.3	60	49	75
Kidney and renal pelvis	1.5	1.9	1.1	67	52	94
Uterus (body, NOS)	—	—	3.1	—	—	32
Thyroid	1.3	0.7	1.9	77	142	53
Pancreas	1.3	1.3	1.3	76	75	79
Oral	1.1	1.5	0.7	89	66	137
Stomach	1.0	1.4	0.7	98	74	142
Multiple myeloma	0.9	1.0	0.7	117	101	137
Brain/CNS	0.6	0.7	0.6	157	140	178
Ovary	—	—	1.3	—	—	75
Esophagus	0.6	0.9	0.3	171	116	314
Liver	0.6	0.9	0.3	168	114	322
Cervix	—	—	0.6	—	—	168
Larynx	0.3	0.4	0.1	389	228	1247
Testis	—	0.4	—	—	236	—
Hodgkin lymphoma	0.2	0.2	0.2	444	411	481

Table A.2: Lifetime probability of developing cancer [1]