# BRAIN TUMOR DETECTION USING MULTIMODAL IMAGE FUSION TECHNIQUE

**BACHELOR OF TECHNOLOGY**

**In**

**ELECTRONICS AND COMMUNICATION ENGINEERING**

**A Major Project Report Submitted by**


**MADIREDDY NARENDRA KUMAR (20011P0413)**

**SAI AMULYA IYYAPU (20011P0421)**

**THANDRA VYSHNAVI (20011P0423)**

**GALITHOTTI NISHANT (20011P0424)**


**Under the esteemed guidance of**


**Dr. M. ASHA RANI**

**Professor in ECE Department**



**Department of Electronics and Communication Engineering**
**Jawaharlal Nehru Technological University Hyderabad – University College of Engineering, Science & Technology Hyderabad-500085**

**2023-2024**

Department of Electronics and Communication Engineering
Jawaharlal Nehru Technological University Hyderabad - University
College of Engineering, Science & Technology Hyderabad - 500085

## CERTIFICATE BY THE SUPERVISOR

**This is to certify that the major project report entitled BRAIN TUMOR DETECTION USING MULTIMODAL IMAGE FUSION TECHNIQUE being submitted by**

**MADIREDDY NARENDRA KUMAR (20011P0413)**

**SAI AMULYA IYYAPU (20011P0421)**

**THANDRA VYSHNAVI (20011P0423)**

**GALITHOTTI NISHANT (20011P0424)**

in partial fulfilment of the requirements for the award of degree in Bachelor of Technology in Electronics and Communication Engineering at the Jawaharlal Nehru Technological University Hyderabad – University College of Engineering, Science & Technology Hyderabad, during the academic year, 2023-24 is a bonafide work carried out under my guidance and supervision. The results embodied in this major project report have been verified and found to be satisfactory.

Supervisor:
Dr. M. Asha Rani
Professor of ECE
JNTUHUCESTH.

**Department Of Electronics and Communication Engineering**

**Jawaharlal Nehru Technological University Hyderabad –**

**University College of Engineering, Science & Technology**

**Hyderabad-500085**

## CERTIFICATE BY THE HEAD OF THE DEPARTMENT

This is to certify that the project report entitled **BRAIN TUMOR DETECTION USING MULTIMODAL IMAGE FUSION TECHNIQUE**

BEING SUBMITTED BY

**MADIREDDY NARENDRA KUMAR (20011P0413)**
**SAI AMULYA IYYAPU (20011P0421)**
**THANDRA VYSHNAVI (20011P0423)**
**GALITHOTTI NISHANT (20011P0424)**

in partial fulfilment of the requirements for the award of degree in Bachelor of Technology in Electronics and Communication Engineering at the Jawaharlal Nehru Technological University Hyderabad – University College of Engineering, Science & Technology Hyderabad, during the academic year, 2023-24 is a bonafide work carried out by them.

Dr.A. Rajani
Professor and Head
Department of ECE, JNTUHUCESTH.

Department Of Electronics and Communication Engineering
Jawaharlal Nehru Technological University Hyderabad-
University College of Engineering, Science & Technology
Hyderabad- 500085

## DECLARATION OF THE CANDIDATES

We hereby declare The Major Project entitled **BRAIN TUMOR DETECTION USING MULTIMODAL IMAGE FUSION TECHNIQUE** is a bonafide record work done and submitted under the esteemed guidance of **Dr.M. Asha Rani**, Professor, Department of ECE, JNTUHUCESTH, in partial fulfillment of the requirements for Major project in Electronics and Communication Engineering at the Jawaharlal Nehru Technological University Hyderabad - University College of Engineering, Science & Technology during the academic year 2023-24 is a bonafide work carried out by us and the results kept in the major project has not been reproduced. The results have not been submitted to any other institute or university for the award of a degree or diploma.

**MADIREDDY NARENDRA KUMAR (20011P0413)**
**SAI AMULYA IYYAPU (20011P0421)**
**THANDRA VYSHNAVI (20011P0423)**
**GALITHOTTI NISHANT (20011P0424)**

## ACKNOWLEDGEMENT

This project titled **BRAIN TUMOR DETECTION USING MULTIMODAL IMAGE FUSION TECHNIQUE** was carried out by us. We are grateful to Prof. Dr. M. Asha Rani and Prof. Dr. A. Rajani, Professor, and Head of the Department of Electronics and Communication Engineering, JNTUHUCESTH, for their guidance while pursuing this project.

We take this precious opportunity to acknowledge our internal project guide Prof. Dr. M.Asha Rani, Professor of Electronics and Communication Engineering, JNTUHUCESTH for her timely advice, effective guidance, and encouragement throughout the completion of our major-project work.

We also owe a deep respect of gratitude to our parents and friends for their cheerful encouragement and valuable suggestions, without whom this work would not have been completed in this stipulated time.

We would like to articulate our heartfelt gratitude to the authorities of JNTUHUCESTH for their help throughout our project work. A few lines of acknowledgment do not fully express our gratitude andappreciation for all those who guided and supported us throughout this project. Lastly, we acknowledge the help received from many journals and websites. Finally, we thank one and all whohelped us directly or indirectly throughout our project work.

**MADIREDDY NARENDRA KUMAR (20011P0413)**
**SAI AMULYA IYYAPU (20011P0421)**
**THANDRA VYSHNAVI (20011P0423)**
**GALITHOTTI NISHANT (20011P0424)**

# INDEX

# LIST OF FIGURES

# LIST OF TABLES

# LIST OF ABBREVIATIONS

| | |
|---|---|
| MRI | Magnetic Resonance Imaging |
| CT | Computed Tomography |
| PET | Positron Emission Tomography |
| SPECT | Single-Photon Emission Computed Tomography |
| VGG19 | Visual Geometry Group 19 |
| GUI | Graphical User Interface |
| CNN | Convolutional Neural Network |
| SVM | Support Vector Machine |
| CNS-pDLBCL | Central Nervous System Primary Diffuse Large B-cell Lymphoma |
| HGG | High-grade Glioma |
| ResNet50 | Residual Network (50-layer architecture) |
| VGG16 | Visual Geometry Group 16 |
| LL | Low-Low |
| LH | Low-High |
| LV | Low-Vertical |
| LD | Low-Diagonal |
| CPU | Central Processing Unit |
| GPU | Graphics Processing Unit |
| CUDA | Compute Unified Device Architecture |
| YCbCr | Luminance-Chrominance-Blue Chrominance-Red Chrominance |
| RGB | Red-Green-Blue |
| IDWT | Inverse Discrete Wavelet Transform |

| | |
|---|---|
| MRF | Markov Random Field |
| BGR | Blue-Green-Red |
| HSV | Hue-Saturation-Value |
| CCA | Connected Component Analysis |
| PIL | Python Imaging Library |
| fMRI | Functional Magnetic Resonance Imaging |

# ABSTRACT

The field of medical imaging has witnessed remarkable advancements in recent years, providing healthcare professionals with powerful tools to diagnose and treat various medical conditions. Among these, brain tumor detection remains a critical area of focus, given the potentially life-threatening nature of such abnormalities. With the advent of multimodal imaging techniques, there has been a paradigm shift in the approach to brain tumor diagnosis, offering a more comprehensive and accurate assessment.

Multimodal imaging involves the integration of information from multiple imaging modalities, such as Magnetic Resonance Imaging (MRI), Computed Tomography (CT) and Positron Emission Tomography (PET). This fusion of data enables a synergistic analysis, harnessing the strengths of each modality to overcome individual limitations. In the context of brain tumor detection, the fusion of diverse imaging modalities holds immense promise, as it can enhance the sensitivity and specificity of diagnostic procedures.

This project focusses on the development and implementation of a Brain Tumor Detection Using Multimodal Imaging Technique. The aim is to expedite diagnosis and treatment initiation, ultimately improving patient outcomes. Moving forward, validation of the methodology using extensive datasets and collaboration with medical professionals to integrate the solution into clinical practice is done.

# CHAPTER-1

# INTRODUCTION

## 1.1 Introduction to Brain Tumor Detection:

A brain tumor is a mass or abnormal growth of cells within the brain or the central spinal canal. These tumors can be benign i.e., non-cancerous or malignant i.e., cancerous, and they can arise from different types of cells within the brain, including neurons, glial cells, and other supportive tissues. Brain tumors can manifest with a variety of symptoms, depending on their size, location, and rate of growth. Common symptoms may include headaches, seizures, cognitive impairment, changes in personality or behaviour, weakness or numbness in the limbs, vision or hearing problems, and nausea or vomiting. However, these symptoms can also be indicative of other neurological conditions, making diagnosis challenging.

The detection of brain tumors typically involves a combination of medical history assessment, neurological examination, and various imaging techniques. Medical history assessment involves gathering information about the patient's symptoms, past medical history, and family history of cancer or genetic disorders. Neurological examination helps assess the patient's cognitive function, motor skills, reflexes, and sensory abilities. Imaging techniques play a crucial role in the detection and diagnosis of brain tumors. Common imaging modalities used include Magnetic Resonance Imaging (MRI), Computed Tomography (CT) scans, and Positron Emission Tomography (PET) scans. MRI is often preferred due to its superior ability to visualize soft tissues, providing detailed images of the brain and helping to differentiate between different types of tumors.

In addition to imaging, other diagnostic tests such as biopsy may be performed to obtain a sample of the tumor tissue for further analysis. This helps determine the tumor's histological type, grade, and genetic characteristics, which are important for treatment planning and prognosis. Early detection of brain tumors is vital for timely intervention and improved outcomes. Advancements in imaging technology, such as functional MRI and PET imaging with radiotracers targeting specific molecular pathways, continue to enhance the accuracy and efficiency of brain tumor detection. Moreover, ongoing research in biomarkers and genetic profiling holds promise for the development of more precise and personalized approaches to brain tumor detection and management.

## 1.2 Overview of Multimodal Imaging Techniques:

Multimodal imaging techniques involve the integration of multiple imaging modalities to provide a more comprehensive assessment of biological tissues or structures. These techniques combine the strengths of different imaging modalities, such as Magnetic Resonance Imaging (MRI), Computed Tomography (CT), Positron Emission Tomography (PET), Single-Photon Emission Computed Tomography (SPECT), and Optical Imaging, to obtain complementary information about physiological processes, anatomy, and molecular characteristics.

Some of the common multimodal imaging techniques are MRI-PET fusion imaging, PET-CT fusion imaging, SPECT-CT fusion imaging, MRI-CT fusion imaging and Multispectral imaging. Of these the MRI-CT fusion imaging is the most preferred for the following reasons:

- **Complementary Information**: MRI and CT offer complementary information about different aspects of the anatomy. MRI excels in soft tissue contrast, making it ideal for visualizing organs, muscles, ligaments, and tumors, while CT provides excellent spatial resolution and bone detail. By combining these modalities, clinicians can obtain a more complete understanding of the patient's anatomy, which is particularly valuable in areas where soft tissues and bony structures coexist, such as the musculoskeletal system.

- **Efficient Workflow**: MRI-CT fusion imaging streamlines the imaging workflow by allowing both modalities to be acquired in a single session, reducing the need for separate imaging studies and subsequent image registration. This enhances efficiency and convenience for both patients and healthcare providers, particularly in time-sensitive clinical scenarios.

- **Enhanced Diagnostic Confidence**: By integrating MRI's superior soft tissue contrast with CT's excellent spatial resolution, MRI-CT fusion imaging enhances diagnostic confidence. It facilitates the identification of subtle abnormalities, improves lesion characterization, and reduces the likelihood of diagnostic errors compared to using either modality alone.

## 1.3 Aim:

Aim of this project is to develop an advanced algorithm and code for precise brain tumor detection using Multimodal Image Fusion Techniques, integrating MRI and CT imaging to create a comprehensive diagnostic tool for identifying subtle abnormalities.

## 1.4 Objectives:

1. Develop precise algorithm for fusing CT and MRI scans for brain tumor detection.
2. Implement landmark registration for accurate spatial alignment of MRI and CT scans.
3. Execute fusion with VGG-19 network for improved spatial and spectral features.
4. Apply watershed segmentation for precise tumor boundary delineation.
5. Craft GUI for seamless execution of all project steps.

## 1.5 Software Requirements:

**Operating System:** Windows

**Language Used:** Python

## 1.6 Methodology:

Brain tumor detection using multimodal image fusion technique is a multi-step process that involves three basic stages Image Registration, Image Fusion and Image Segmentation. Below is a detailed methodology for implementing such algorithm:

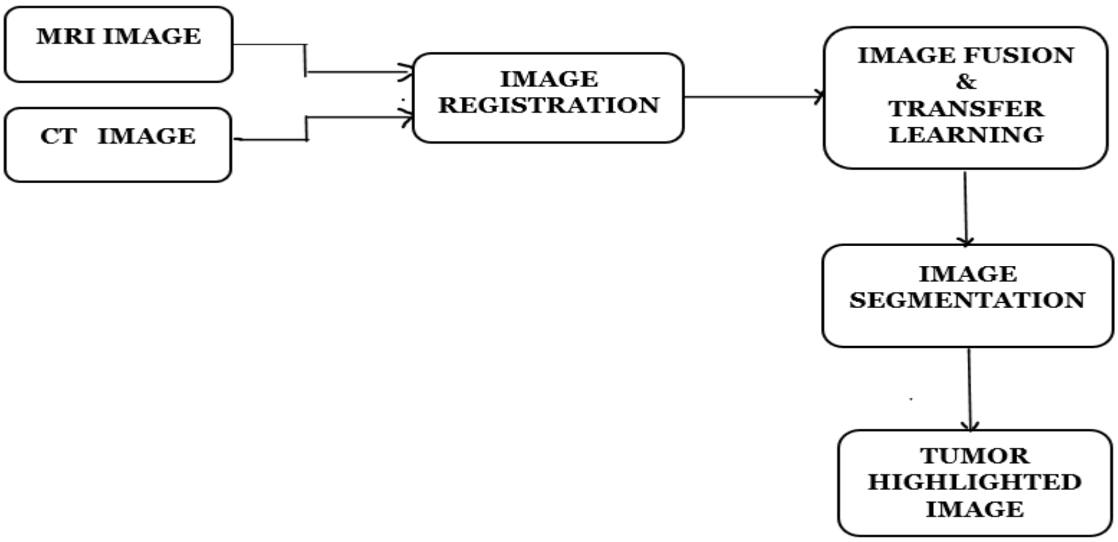


*Figure 1.1 Tumor Detection Block Diagram*

### 1.6.1Input Images:

- The input to the algorithm is two sets of images, MRI and CT scan images of a single patient. To get these images first a referral from a qualified healthcare provider such as a physician or specialist has to be obtained. Then the scan can be undergone at a registered imaging facility. Once the results are received, they can be interpreted by a radiologist to know the condition of the patient's body.

### 1.6.2 Image Registration:

- Image registration is a fundamental process in medical imaging that involves aligning and merging multiple images of the same subject or anatomical area acquired from different imaging modalities, at different times, or from different perspectives. The goal of image registration is to spatially align these images so that corresponding features or structures overlap accurately, facilitating comparison, analysis, and integration of information from various sources.

### 1.6.3 Image Fusion and Transfer Learning:

- Image fusion is a process that involves combining multiple images of the same scene or subject acquired from different sources or sensors to create a single composite image that contains more information than any of the individual input images. The goal of image fusion is to enhance the visibility of relevant features, improve image quality, and provide a more comprehensive representation of the scene.
- Transfer learning is a machine learning technique where knowledge gained from training a model on one task or dataset is transferred and applied to a related but different task or dataset. Instead of starting the learning process from scratch, transfer learning leverages pre-trained models or learned representations to bootstrap the learning process on a new task, typically with fewer labeled examples or training data.

### 1.6.4 Image Segmentation:

- Image segmentation is a fundamental task in image processing and computer vision that involves partitioning an image into multiple meaningful and homogeneous regions or segments. The goal of image segmentation is to divide an image into distinct regions based on certain criteria, such as color, intensity, texture, or spatial proximity, to extract and delineate objects or regions of interest within the image.

### 1.6.5 Tumor Highlighted Image:

- This is the output of the algorithm which is an image that represents the regions of the brain which are infected due to tumor or any other similar abnormalities. This image will be helpful for doctors for further analysis for doctors to know the position of the brain tumor and carrying out further treatments if required.

Each of these blocks i.e., Image Registration, Image Fusion and Image Segmentation are explained in detail in the subsequent chapters.

# CHAPTER - 2
# LITERATURE REVIEW

Brain tumor detection is a critical task in medical imaging, as early diagnosis significantly impacts treatment outcomes and patient survival rates. Magnetic Resonance Imaging (MRI) is widely utilized for its superior image quality in detecting brain tumors. However, traditional methods can be time-consuming and may lack accuracy. Recent advancements in deep learning techniques have shown promise in improving both accuracy and efficiency in brain tumor detection. This literature review aims to explore various approaches proposed in recent research papers to detect brain tumors using MRI images and their potential implications for a project on Multimodal Medical Image Fusion for Brain Tumor Detection.

**“Enhancing Brain Tumor Detection on MRI Images Using an Innovative VGG-19 Model-Based Approach”:**

Abdullah SENER and Burhan ERGEN proposed a novel approach utilizing the VGG-19 convolutional neural network (CNN) model for accurate brain tumor detection on MRI images. By leveraging deep learning techniques, their model achieved high accuracy and demonstrated promising potential for clinical applications. The study underscores the significance of early detection in improving patient prognosis, emphasizing the role of advanced technologies in assisting healthcare professionals in making informed treatment decisions.

**“On the Performance of Deep Transfer Learning Networks for Brain Tumor Detection Using MR Images”:**

Saif Ahmad and Pallab K. Choudhury conducted an in-depth analysis of deep transfer learning methods for brain tumor detection, utilizing seven different pre-trained models and traditional classifiers. Their study showcased the superior performance of the VGG-19-SVM model, achieving an impressive accuracy of 99.39%. The research highlights the efficacy of transfer learning in enhancing brain tumor detection accuracy and calls for further exploration to optimize model performance.

**“Image Classification of Brain Tumor Based on Enhanced VGG 19 CNN”:**

Tian C and Zhao M study introduces an enhanced VGG19 CNN model for classifying primary central nervous system diffuse large B-cell lymphoma (CNS-pDLBCL) and high-grade glioma (HGG) based on MRI images. By incorporating transfer learning and support vector machine classification, the proposed model demonstrated superior accuracy compared to existing methods. The research emphasizes the importance of objective and accurate pathological diagnosis facilitated by advanced deep learning techniques.

**“Image Analysis for MRI-Based Brain Tumor Classification Using Deep Learning”:**

Krisna Nuresa Qodri, Indah Soesanti, and Hanung Adi Nugroho explored the application of deep learning and transfer learning techniques for MRI-based brain tumor classification. Their study evaluated various models, including ResNet50 and VGG16, and reported high accuracy rates for both models. The research underscores the potential of transfer learning in handling medical image data and emphasizes the need for further investigation with larger datasets to enhance model generalization.

**“Detection of Brain Tumor Using Digital Image Processing”:**

Sheeba Khan’s research focuses on brain tumor detection using digital image processing techniques, particularly thresholding, on MRI images. While not employing deep learning methods, the study presents an alternative approach for tumor detection, emphasizing simplicity and efficiency. The proposed GUI-based tumor detection system holds promise for widespread use, offering users the ability to detect and measure tumor size accurately.

*Deep learning, especially CNNs, shown promising results for enhancing brain tumor detection accuracy on MRI images. Transfer learning, like fine-tuning pre-trained models such as VGG-19, consistently yields high accuracy rates. Integrating modalities like CT scans may improve diagnostic accuracy.*

*The motivation for this project is to leverage cutting-edge and image fusion techniques to improve brain tumor detection, significantly impacting patient care and advancing medical technology. The project aims to address gaps in current methods by employing advanced fusion techniques and ensuring precise spatial alignment, potentially impacting treatment planning and patient outcomes.*

# CHAPTER - 03

# Image Registration

## 3.1 Introduction to Image Registration:

Image registration is a fundamental process in medical imaging that involves aligning two or more images of the same subject to a common coordinate system. This alignment enables the integration and comparison of information from different imaging modalities, such as MRI and CT scans, enhancing the accuracy and completeness of diagnostic information. Image registration is essential for various medical applications, including tumor detection, treatment planning, and monitoring of disease progression. The process typically involves extracting features or landmarks from the images, optimizing a spatial transformation to align the features, and validating the accuracy of registration. By aligning images, image registration facilitates the visualization, analysis, and interpretation of complex anatomical structures and pathological conditions, ultimately improving patient care and outcomes.

Overall, the adoption of image registration in medical imaging is driven by its ability to enhance diagnostic accuracy, improve treatment outcomes, and streamline clinical workflows. Image registration in the context of aligning MRI and CT scans involves the process of spatially aligning the two different imaging modalities, MRI and CT, so that corresponding anatomical structures in the images are in the same spatial locations.

The detailed procedure for image registration is explained in the following sections.

## 3.2 Working Principle: Following are the steps for Image Registration process:

1. **Image Acquisition:** MRI and CT scans provide different types of information about the same anatomical structures. MRI is known for its excellent soft tissue contrast, while CT provides good contrast for bones and dense tissues. Both modalities are commonly used in medical imaging for diagnosing conditions such as brain tumors.

2. **Preprocessing:** Before registration, both MRI and CT images may undergo preprocessing steps to enhance their quality and reduce noise or artifacts. Preprocessing may include filtering, intensity normalization, and geometric correction.

3. **Feature Extraction:** Features such as key points, edges, or anatomical landmarks are extracted from both the MRI and CT images. These features serve as reference points for aligning the images.

**4. Registration:** The registration process involves finding a spatial transformation that maps the features from one image to corresponding features in the other image. Various registration techniques can be used, including rigid, affine, or non-rigid transformations. Rigid registration preserves angles and distances, while affine registration allows for scaling, rotation, and translation. Non-rigid registration allows for more complex deformations.

**5. Optimization:** The spatial transformation parameters are optimized to minimize the difference between corresponding features in the MRI and CT images. Optimization algorithms such as gradient descent or genetic algorithms may be used to find the optimal transformation.

**6. Integration and Analysis:** Once the MRI and CT images are aligned, they can be integrated or fused into a single representation. This integrated image provides comprehensive information about the anatomy and pathology of the patient, facilitating diagnosis, treatment planning, and monitoring of conditions such as brain tumors.

Overall, image registration plays a crucial role in aligning MRI and CT scans, enabling the integration of complementary information from these modalities for improved diagnosis and treatment of medical conditions.

## 3.3 Landmark Registration:

Landmark registration, a subset of image registration, focuses on aligning images based on identifiable landmarks or points of interest within the images. In medical imaging, landmark registration involves identifying anatomical landmarks, such as specific bones, vessels, or tissue boundaries, and using them as reference points for alignment. Landmark registration offers several advantages, including simplicity, robustness, and interpretability. By selecting prominent anatomical features as landmarks, registration algorithms can achieve accurate alignment even in the presence of image distortions, noise, or variations in image contrast.

For brain tumor detection using CT and MRI scans, landmark registration can be a pivotal step in aligning the two imaging modalities. Landmark registration involves identifying distinctive anatomical landmarks within the brain region that are discernible in both the CT and MRI scans. These landmarks serve as reference points to ensure accurate spatial alignment between the two modalities. By pinpointing consistent and easily recognizable landmarks, such as key structures along the skull base or specific points within the ventricular system, landmark registration facilitates precise alignment despite differences in imaging characteristics between CT and MRI. This alignment is crucial for enabling comprehensive analysis and comparison of the images, ultimately enhancing the accuracy and reliability of brain tumor detection algorithms.

The landmark registration involves capturing specific points of interest called landmarks, from both the CT and MRI images and using them as reference points for alignment.

Below is the detailed procedure how the code achieves this Landmark Registration:

**1. Capturing Landmarks:** The code defines two mouse callback functions, 'click_event_ct'and 'click_event_mri', which capture mouse clicks on the displayed CT and MRI images, respectively. These functions trigger when the left mouse button is clicked ('EVENT_LBUTTONDOWN'). Upon a click event, the '(x, y)' coordinates of the click are printed to the console and appended to the 'ct_points' or 'mri_points' lists, depending on the image being clicked.

**2. User Interaction:** The user interacts with the displayed CT and MRI images by clicking on specific anatomical landmarks visible in both images. These landmarks can include distinct features within the brain region relevant to tumor detection, such as points along the skull base or specific locations within the ventricular system.

**3. Recording Landmarks:** As the user clicks on landmarks in the images, their coordinates are recorded in the 'ct_points' and 'mri_points' lists, respectively. Each list contains the '(x, y)' coordinates of the clicked points, representing the identified landmarks in the CT and MRI images.

**4. Utilizing Landmarks for Alignment:** Once the landmarks are captured from both images, they can be utilized in a landmark-based registration algorithm to compute the spatial transformation necessary to align the images. This transformation aims to bring the identified landmarks into correspondence, ensuring accurate alignment of the entire images.

**5. Enhancing Registration Accuracy:** By using anatomically meaningful landmarks for alignment, the registration process becomes more robust and accurate, particularly in cases of anatomical variability or image artifacts. Landmark registration facilitates precise spatial alignment between the CT and MRI images, enabling more reliable analysis and interpretation for brain tumor detection.

- To implement Procrustes Analysis is used to align shapes or configurations of points by minimizing differences between them. In landmark registration, it aligns two sets of landmarks from different images by finding the optimal transformation (translation, rotation, and scaling) that minimizes their differences. In this algorithm, Procrustes Analysis aligns manually selected landmarks on CT as shown in figure 3.2 and MRI images as shown in figure 3.1, registering them spatially as in figure 3.3 for further analysis and comparison.

## 3.4 Libraries Used:

- **NumPy (np):** NumPy is a fundamental package for numerical computing with Python. It provides support for arrays, matrices, and mathematical functions to operate on these arrays.

- **matplotlib.pyplot (plt)**: this module is a powerful plotting library for Python. It provides a MATLAB-like interface for creating and customizing various types of plots and visualizations in Python. It is widely used for creating graphs in scientific computing, data analysis, and visualization tasks.

- **matplotlib.cm (cm):** this module cm from the matplotlib library, allows the access of various colormaps for visualizing data in plots and images.
- **OpenCV (cv2)**: This library is used for image processing tasks such as reading, writing, and manipulating images.
- **imageio:** This library is used for reading and writing a wide range of image data. It provides an easy-to-use interface for working with images in different formats.
- **scipy.ndimage (ndi):** This module contains various functions for multi-dimensional image processing. It includes functions for filtering, interpolation, morphological operations, and other image manipulation tasks.

## 3.4 Implementation Details:

The code facilitates image registration through the following steps:

**1. Interactive Landmark Selection:** Users select landmarks on both the CT and MRI images by clicking on specific points of interest. This is facilitated by the `click_event_ct` and `click_event_mri` functions.

```
# Define Click Function
def click_event_mri(event, x, y, flags, param):
    if event == cv2.EVENT_LBUTTONDOWN:
        print(x,y)
        mri_points.append([x,y])
```

```
# Define Click Function
def click_event_ct(event, x, y, flags, param):
    if event == cv2.EVENT_LBUTTONDOWN:
        print(x,y)
        ct_points.append([x,y])
```

**2. Conversion to NumPy Arrays:** The coordinates of the selected landmarks are converted into NumPy arrays (‘X_pts’ and ‘Y_pts’) for further processing.

```
X_pts = np.asarray(ct_points)
Y_pts = np.asarray(mri_points)
print(X_pts)
```

**3. Procrustes Analysis:** The 'procrustes' function is applied to the landmark coordinates from both images ('X_pts' and 'Y_pts'). Procrustes analysis calculates the optimal transformation (rotation, scaling, and translation) to align the landmarks from one set to the other, minimizing the sum of squared differences between corresponding points.

```
def procrustes(X, Y, scaling=True, reflection='best'):

    n,m = X.shape
    ny,my = Y.shape

    muX = X.mean(0)
    muY = Y.mean(0)

    X0 = X - muX
    Y0 = Y - muY

    ssX = (X0**2.).sum()
    ssY = (Y0**2.).sum()
    print(ssX)
    print(ssY)

    # centred Frobenius norm
    normX = np.sqrt(ssX)
    normY = np.sqrt(ssY)

    # scale to equal (unit) norm
    X0 /= normX
    Y0 /= normY
    if my < m:
        Y0 = np.concatenate((Y0, np.zeros(n, m-my)),0)

    # optimum rotation matrix of Y
    A = np.dot(X0.T, Y0)
    U,s,Vt = np.linalg.svd(A,full_matrices=False)
    V = Vt.T
    T = np.dot(V, U.T)
```

```python
if reflection is not 'best':

    # does the current solution use a reflection?
    have_reflection = np.linalg.det(T) < 0

    # if that's not what was specified, force another reflection
    if reflection != have_reflection:
        V[:,-1] *= -1
        s[-1] *= -1
        T = np.dot(V, U.T)

traceTA = s.sum()

if scaling:

    # optimum scaling of Y
    b = traceTA * normX / normY

    # standarised distance between X and b*Y*T + c
    d = 1 - traceTA**2
    # transformed coords
    Z = normX*traceTA*np.dot(Y0, T) + muX

else:
    b = 1
    d = 1 + ssY/ssX - 2 * traceTA * normY / normX
    Z = normY*np.dot(Y0, T) + muX
```

```python
# transformation matrix
if my < m:
     T = T[:my,:]
c = muX - b*np.dot(muY, T)
#rot =1
#scale=2
#translate=3
#transformation values
tform = {'rotation':T, 'scale':b, 'translation':c}

return d, Z, tform
```

**4. Transformation Computation:** The resulting transformation parameters ('Tform') are used to compute the transformation matrix 'M', which aligns the MRI image with the CT image.

```python
d,Z_pts,Tform = procrustes(X_pts,Y_pts)
```

**5. Image Transformation:** The computed transformation matrix 'M' is applied to the MRI image ('mri_registered') using the 'cv2.warpAffine' function, resulting in the aligned MRI image ('tr_Y_img').

```
S = np.eye(3) * Tform['scale']
S[2,2] = 1
t = np.eye(3)
t[0:2,2] = Tform['translation']
M = np.dot(np.dot(R,S),t.T).T
h=ct.shape[0]
w=ct.shape[1]
tr_Y_img = cv2.warpAffine(mri_registered,M[0:2,:],(h,w))
cv2.imwrite("jpg/mri_registered.jpg", tr_Y_img)
```

## 3.4 Registered Images:

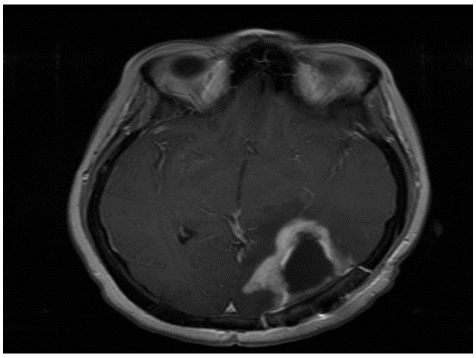

*Figure 3.1 MRI Image of Patient*

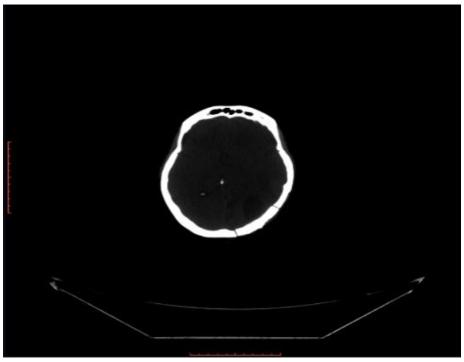

*Figure 3.2 CT Image of Patient*

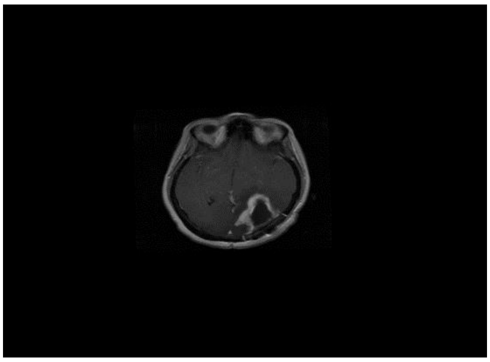

*Figure 3.3 Registered MRI of Patient w.r.t to patient's CT Image*

# CHAPTER-04

# IMAGE FUSION

## 4.1 Introduction to Image Fusion:

Image fusion integrates information from multiple images to create a single, enhanced composite image, used across fields like medicine and remote sensing. By combining data from different sources, it improves image quality, resolution, and interpretability, aiding in tasks such as surveillance and computer vision. It addresses limitations like noise and low contrast in individual images, resulting in clearer representations. Various fusion methods exist, catering to specific application needs, from pixel-level to decision-level fusion. Overall, image fusion facilitates better image understanding, decision-making, and visualization in diverse scenarios.

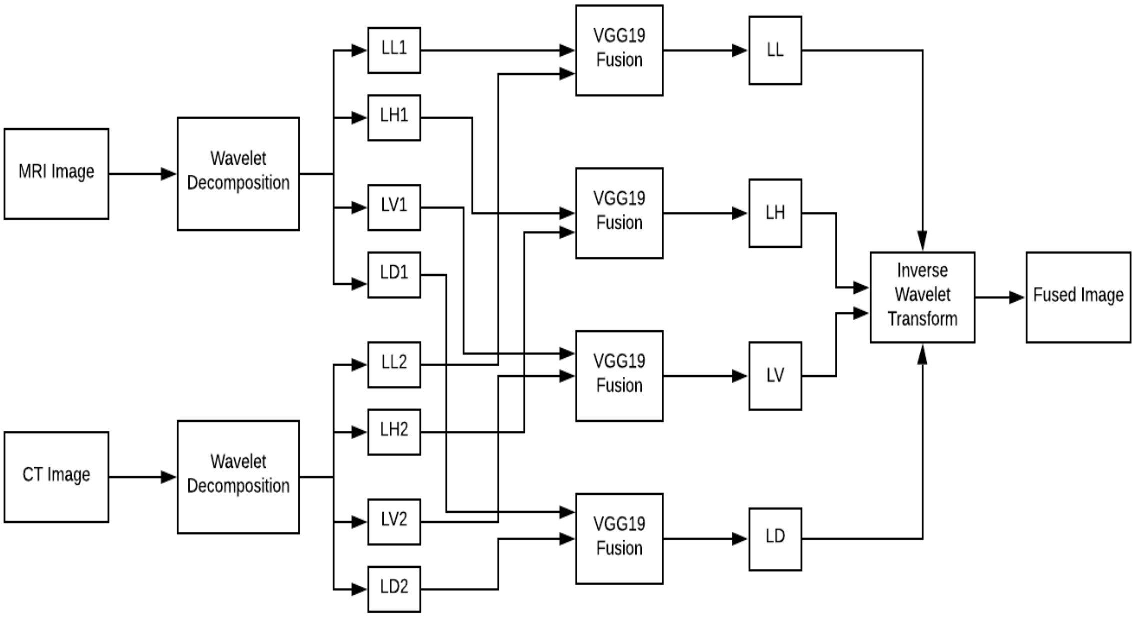


*Figure 4.1 Process of Image Fusion*

## 4.2 Wavelet Transform and Image Decomposition:

Wavelet transform is a mathematical technique used for signal and image processing. Unlike traditional Fourier transform, which decomposes signals into sinusoidal components of different frequencies, wavelet transform decomposes signals into a set of basic functions called wavelets. These wavelets are localized in both time and frequency domains, which allows for capturing both high-frequency and low-frequency components of a signal or an image efficiently.

Wavelet transform operates by analyzing the signal or image at different scales and resolutions. It performs a multi-resolution analysis by decomposing the signal or image into approximation coefficients (representing low-frequency components) and detail coefficients (representing high-frequency components) at each scale level.

In image processing, wavelet transform is commonly used for tasks such as denoising, compression, feature extraction, and image fusion.

## 4.3 Implementation Details for Image Decomposition:

We start by loading the MRI and CT images using the OpenCV library and converting them to grayscale to simplify the analysis. The wavelet transform is then applied using the PyWavelets library, specifically the pywt.dwt2() function with the 'haar' wavelet, known for its simplicity and efficiency. This transformation breaks down each image into four components: approximation (LL), horizontal detail (LH), vertical detail (HL), and diagonal detail (HH) as in figure 4.2 for MRI Image decomposition and figure 4.3 for CT Image decomposition.

The resulting decomposition is visualized through a series of sub-images, each representing a distinct aspect of the original image. The approximation component captures the overall structure and smooth regions of the image, providing a low-frequency representation. Meanwhile, the detail components highlight specific features such as horizontal, vertical, and diagonal edges, revealing high-frequency information.

**MRI Image Decomposition:**

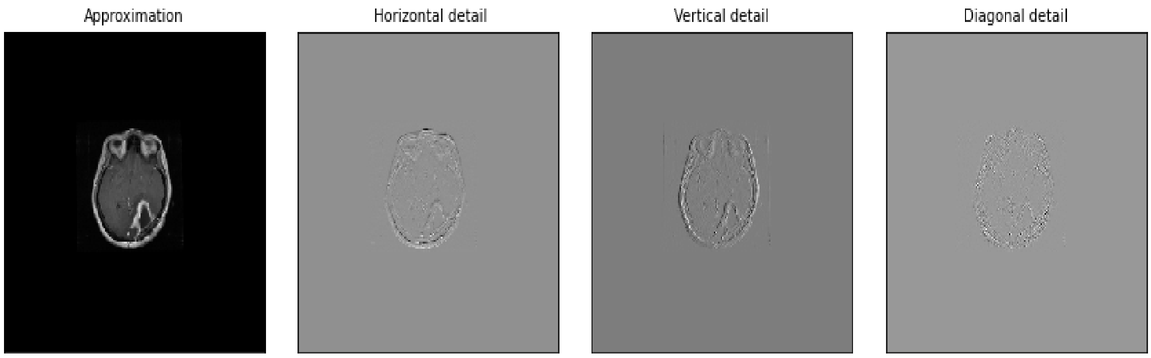


*Figure 4.2 MRI Image Decomposition into Approximation, Horizontal detail, Vertical detail and Diagonal detail Images*

**CT Image Decomposition:**

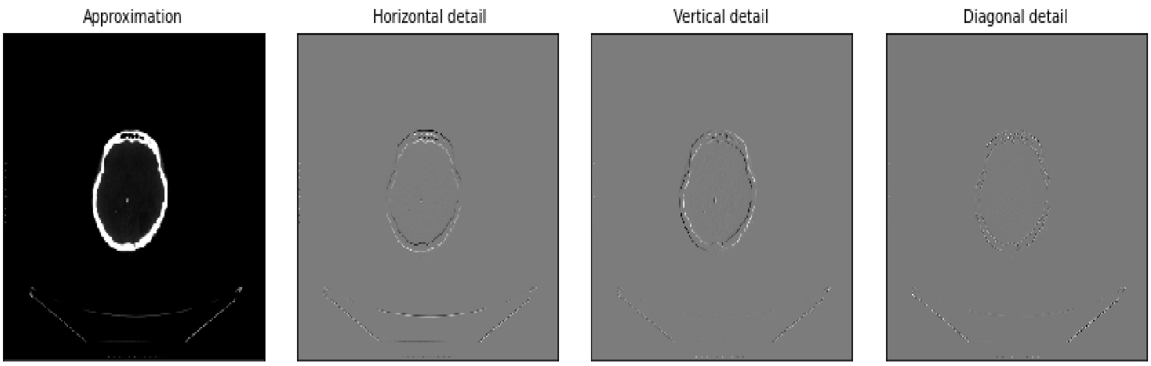


*Figure 4.3 CT Image Decomposition into Approximation, Horizontal detail, Vertical detail and Diagonal detail Images*

By examining the wavelet decomposition of both MRI and CT images, we gain insights into their unique characteristics and underlying structures. Differences in the decomposition patterns may indicate variations in image texture, contrast, or structural complexity between the two modalities. Such insights are valuable for tasks such as image analysis, segmentation, and feature extraction.

## 4.4 VGG19 Model for Image Feature Extraction:

VGG19 is a deep convolutional neural network architecture proposed by the Visual Geometry Group (VGG) at the University of Oxford. It consists of 19 layers (hence the name) including convolutional layers, max-pooling layers, and fully connected layers. VGG19 is widely used for various computer vision tasks including image classification, object detection, and feature extraction. In the context of image processing, VGG19 can be used as a feature extractor. The network is pre-trained on large-scale image datasets (e.g., ImageNet) to learn hierarchical features at different levels of abstraction. By removing the fully connected layers of VGG19, the network can be used to extract features from images, which can then be used for tasks such as image fusion, style transfer, etc.

We initialized the VGG19 model for feature extraction. It loads the pre-trained weights and defines a forward pass method to extract intermediate feature maps from input images. This modular approach allows for easy integration of feature extraction capabilities into different deep learning pipelines.

## 4.5 Libraries Used:

- **argparse**: This is a standard library for parsing command-line arguments.
- **OpenCV (cv2)**: This library is used for image processing tasks such as reading, writing, and manipulating images.
- **NumPy (np)**: NumPy is a fundamental package for numerical computing with Python. It provides support for arrays, matrices, and mathematical functions to operate on these arrays.
- **PyTorch**: PyTorch is an open-source machine learning library. We need both the torch and torch.nn modules.
- **torch.nn.functional (F)**: This module provides functions that operate on input tensors.
- **torchvision**: It is a PyTorch library that provides utilities and datasets for vision tasks.
- **Pywt:** It is a Python library that provides functions and classes for performing wavelet transform and related signal processing tasks.

## 4.6 Image Fusion Implementation:

Image fusion integrates information from multiple images to create a single, enhanced composite image, used across fields like medicine and remote sensing. By combining data from different sources, it improves image quality, resolution, and interpretability, aiding in tasks such as surveillance and computer vision. It addresses limitations like noise and low contrast in individual images, resulting in clearer representations. Various fusion methods exist, catering to specific application needs, from pixel-level to decision-level fusion. Overall, image fusion facilitates better image understanding, decision-making, and visualization in diverse scenarios.

The Fusion class is designed to handle the process of image fusion, which involves combining multiple input images into a single output image while preserving relevant information from each input. Here is a detailed explanation of its components:

**Initialization:**

Upon instantiation, the Fusion class receives a set of input images. It determines whether to utilize CPU or GPU for computation based on the availability of CUDA. Additionally, it initializes an instance of the VGG19 model, a convolutional neural network (CNN), which will be used to extract features from the input images.

**Image Fusion Method (fuse()):**

The fuse() method orchestrates the image fusion process. Here is a breakdown of its steps:

- **Color Space Conversion**:

  The method begins by converting all input images to the YCbCr color space. This conversion is performed to facilitate subsequent processing steps.

- **Normalization:**

  Next, the images are normalized to ensure consistent data range for further computations. For non-grayscale images, only the luminance component (Y channel) is retained.

- **Conversion to PyTorch Tensors:**

  The normalized images are then transferred to PyTorch tensors. This step is necessary to prepare the images for processing with the VGG19 model.

- **Fusion Algorithm Execution:**

  The method executes the fusion algorithm, which involves extracting feature maps from the input images using the VGG19 model. These feature maps are then combined using a weighted sum approach, where the weights are determined based on the softmax output of the feature maps. The fused image is generated as the output of this fusion process.

- **Reconstruction:**

  Finally, the fused image is reconstructed in RGB format. This involves converting the YCbCr image back to RGB and clipping any out-of-range pixel values.

**Private Fusion Algorithm (_fuse()):**

The _fuse() method contains the core fusion algorithm. It iteratively processes the feature maps obtained from input images, computes their weighted sum, and generates the fused image based on the calculated weights.

**Utility Methods:**

The Fusion class also includes several utility methods for internal use:

- **_RGB_to_YCbCr() and _YCbCr_to_RGB():** These methods handle the conversion between RGB and YCbCr color spaces.
- **_is_gray():** Determines whether an image is grayscale based on its channel dimensions.
- **_softmax():** Computes the softmax output of a tensor, which is used to calculate weights in the fusion algorithm.
- **_tranfer_to_tensor():** Transfers input images to PyTorch tensors, ensuring compatibility with the VGG19 model.

The Fusion class provides a comprehensive framework for performing image fusion using deep learning techniques. By encapsulating the fusion process and associated functionalities, it offers a modular and efficient solution for combining input images while preserving relevant information.

## 4.7 Implementation of VGG19 for Multispectral Image Fusion:

In this implementation, we utilize the VGG19 neural network architecture for the task of multispectral image fusion. Multispectral image fusion involves combining information from multiple spectral bands to produce a single fused image with enhanced quality and information content.

**Methodology:**

- Data Preparation: We obtained four bands (LL, LH, LV, LD) from wavelet-transformed MRI and CT images.
- VGG19 Initialization: The pretrained VGG19 model was configured to extract features from each band, capturing both semantic and spatial information.
- Feature Extraction: Features were extracted from the third convolutional layer of VGG19, striking a balance between detail preservation and semantic understanding.
- Fusion Algorithm: We computed softmax weights for each feature map and combined them to produce the final fused image.
- Image Fusion: Each band combination (LL, LH, LV, LD) underwent fusion separately, with input images passed through VGG19 for feature extraction.

Fused images were saved as JPEG files using OpenCV for further analysis as in figure 4.4 and 4.5. Fused images showed improved quality and detail compared to individual bands, particularly in regions with complementary information.

## 4.8 Reconstruction Using Inverse Wavelet Transform:

- **Loading Fused Images:**

The fused images generated from the fusion process are loaded using OpenCV's cv2.imread() function.

Each image is converted to grayscale using cv2.cvtColor() function, as the wavelet transform typically operates on single-channel images.

- **Constructing Coefficients Tuple:**

The grayscale fused images are then organized into a tuple co-efficients that represents the coefficients needed for the inverse wavelet transform.

Each fused image corresponds to one detail coefficient (fusion_0) or three approximation coefficients (fusion_1, fusion_2, fusion_3).

- **Inverse Wavelet Transform:**

The pywt.idwt2() function is used to perform the inverse discrete wavelet transform (IDWT) on the co-efficients tuple.

The 'haar' wavelet is specified as the type of wavelet to be used for the reconstruction.

- **Saving the Final Fusion Image:**

The resulting fused image from the IDWT process is saved using cv2.imwrite() function.

## 4.9 Fused Images of Various Patients:

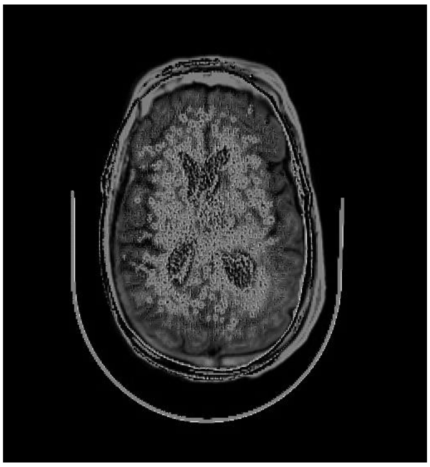

*Figure 4.4 Fused Image of Patient-1*

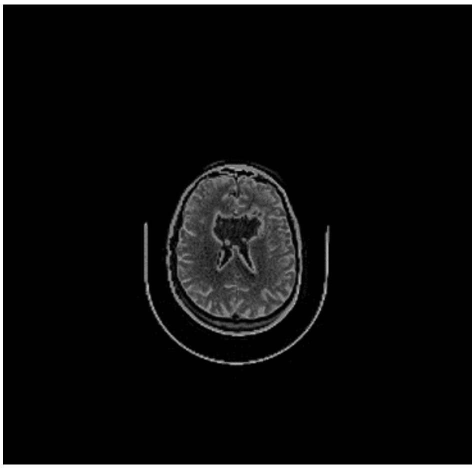

*Figure 4.5 Fused Image of Patient-2*

# CHAPTER-5

# IMAGE SEGMENTATION

## 5.1 Introduction to Image Segmentation:

Image segmentation is a pivotal task in the realm of computer vision and image processing, serving as a foundational technique for extracting meaningful information from digital images. At its core, image segmentation involves partitioning an image into distinct regions or segments based on certain characteristics, such as color, intensity, texture, or spatial proximity. This process enables the identification and delineation of objects, boundaries, or regions of interest within an image, laying the groundwork for subsequent analysis and interpretation.

## 5.2 Watershed Segmentation:

Watershed segmentation is a popular image segmentation technique inspired by the concept of hydrology, where an image is treated as a topographic surface, and segmentation is performed by simulating flooding from local minima. This method is particularly useful for segmenting objects with irregular shapes and poorly defined boundaries.

**Working of Watershed Segmentation:**

1. **Gradient Calculation**: The first step in watershed segmentation involves computing the gradient of the image to highlight regions of significant intensity variation. This gradient image represents the strength of edges or boundaries in the original image.

2. **Marker Selection**: Local minima in the gradient image are identified as potential "seeds" or markers for segmentation. These markers serve as starting points for the flooding process and help guide the segmentation algorithm.

3. **Flood-filling**: Starting from the markers, a flooding process is initiated to partition the image into different segments. The flooding process simulates the filling of basins in the topographic surface, where water flows from each marker and floods neighbouring pixels until boundaries are encountered.

4. **Watershed Transformation**: As flooding progresses, boundaries between adjacent basins or regions are delineated where flooding fronts meet. These boundaries are known as watershed lines and represent potential segment boundaries in the image. Watershed lines are formed where flooding basins merge and form catchment basins, analogous to valleys in a topographic map.

5. **Segmentation Result**: The final segmentation result as shown in figures 5.1 and 5.2, is obtained by assigning each pixel in the image to the nearest catchment basin or watershed region. Pixels within the same catchment basin belong to the same segment or region, while pixels separated by watershed lines belong to different segments.

Watershed segmentation is preferred over other segmentation algorithms in this application due to the following advantages:

1. **Boundary Preservation**: Watershed segmentation tends to produce segmentations with well-defined boundaries, making it particularly suitable for segmenting anatomical structures or objects with irregular shapes and complex boundaries, such as tumors in medical images. This boundary preservation property ensures accurate delineation of structures of interest, which is critical for subsequent analysis and diagnosis.

2. **Edge Sensitivity**: Watershed segmentation is sensitive to gradient variations in the image, which makes it effective in detecting subtle intensity changes or edges between different tissue types or structures. This sensitivity is beneficial for segmenting structures with distinct boundaries, such as blood vessels, organs, or lesions, in medical images acquired from modalities like MRI, CT, or ultrasound.

3. **Flexibility and Adaptability**: Watershed segmentation is versatile and adaptable to various imaging modalities and image characteristics. It can handle images with different resolutions, noise levels, and intensity distributions, making it robust and widely applicable across different medical imaging applications. Additionally, watershed segmentation can be combined with other segmentation techniques or preprocessing steps to further enhance segmentation accuracy and robustness.

4. **Segmentation Quality**: Despite its susceptibility to over-segmentation in some cases, watershed segmentation generally produces high-quality segmentations with minimal artifacts or errors, especially when appropriate markers are used and post-processing techniques are applied. This segmentation quality is crucial for reliable quantitative analysis, diagnosis, and treatment planning in medical imaging.

## 5.3 Libraries Used:

- **OpenCV (cv2)**: This library is used for image processing tasks such as reading, writing, and manipulating images.

- **NumPy (np)**: NumPy is a fundamental package for numerical computing with Python. It provides support for arrays, matrices, and mathematical functions to operate on these arrays.

- **matplotlib.pyplot (plt)**: this module is a powerful plotting library for Python. It provides a MATLAB-like interface for creating and customizing various types of plots and visualizations in Python. It is widely used for creating graphs in scientific computing, data analysis, and visualization tasks.

- **Skimage.morphology.extrema**: The extrema module from the skimage.morphology package in scikit-image provides functions for finding regional extrema (local minima and maxima) in images, which can serve as markers for the algorithm.
- **Skimage.morphology.watershed (skwater):** The watershed module from the skimage.morphology package in scikit-image provides an implementation of the watershed algorithm for image segmentation. The watershed algorithm is a powerful technique for segmenting objects in an image based on the gradient or intensity of the image.

## 5.4 Watershed Segmentation Implementation:

To implement watershed segmentation for our application the following steps have to be followed sequentially:

1. **Importing Libraries:** You import necessary libraries such as NumPy for numerical operations, OpenCV for image processing, matplotlib for visualization, and scikit-image for morphology operations and watershed algorithm.
2. **Definition of ShowImage Function:** This function is defined to display images with different color types (BGR, HSV, grayscale, or RGB).

```
def ShowImage(title,img,ctype):
    plt.figure(figsize=(10, 10))
    if ctype=='bgr':
        b,g,r = cv2.split(img)       # get b,g,r
        rgb_img = cv2.merge([r,g,b])     # switch it to rgb
        plt.imshow(rgb_img)
    elif ctype=='hsv':
        rgb = cv2.cvtColor(img,cv2.COLOR_HSV2RGB)
        plt.imshow(rgb)
    elif ctype=='gray':
        plt.imshow(img,cmap='gray')
    elif ctype=='rgb':
        plt.imshow(img)
    else:
        raise Exception("Unknown colour type")
    plt.axis('off')
    plt.title(title)
    plt.show()
```

The function consists of three parameters:

- title: The title to be displayed above the image.
- img: The image data to be displayed.
- ctype: The color type of the image ('bgr', 'hsv', 'gray', or 'rgb').

Based on the **ctype** parameter, the function handles different color types of images:

- If **ctype** is 'bgr', it assumes the image is in BGR (Blue-Green-Red) color format and splits the channels and reorders them to RGB format before displaying.
- If **ctype** is 'hsv', it assumes the image is in HSV (Hue-Saturation-Value) color format and converts it to RGB before displaying.
- If **ctype** is 'gray', it assumes the image is grayscale and displays it using a grayscale colormap.
- If **ctype** is 'rgb', it assumes the image is already in RGB format and displays it as is.

The function then displays the image using Matplotlib's **imshow** function, along with the specified title. The axis is turned off to remove the axis labels, and finally, the image is shown using **plt.show()**.

This function provides a convenient way to display images with different color formats and types using Matplotlib.

3. **Loading and Preprocessing the Image**: The input image ("final_fusion.jpg") is loaded using OpenCV's **cv2.imread** function. Then, the image is converted to grayscale using **cv2.cvtColor** function.
4. **Thresholding**: Otsu's thresholding method is applied to the grayscale image to obtain a binary thresholded image.
5. **Connected Component Analysis (CCA)**: Connected components are extracted from the thresholded image using **cv2.connectedComponents**. This step is crucial for identifying the brain region in the image.
6. **Brain Mask Generation**: The largest connected component (assumed to be the brain region) is extracted, and a binary mask representing the brain region is created.
7. **Brain Segmentation**: The brain region is extracted from the original image by applying the binary mask.
8. **Additional Preprocessing**: The input image is reloaded, and additional preprocessing steps are performed to prepare it for watershed segmentation. These steps include thresholding, noise removal using morphological opening, distance transformation, and determination of sure background and sure foreground areas.
9. **Watershed Algorithm**: The watershed algorithm is applied to segment the image into distinct regions. The segmented regions are labelled using connected components, and the boundaries between regions are marked with -1.
10. **Visualization and Saving the Segmented Image**: The segmented image is visualized using the **ShowImage** function and saved to a file ("segmented.jpg").

The above steps from 3-10 can be implemented using the below code:

```
img = cv2.imread("jpg/final_fusion.jpg")
gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
ShowImage('Image',gray,'gray')

ret, thresh = cv2.threshold(gray,0,255,cv2.THRESH_OTSU)
ShowImage('Thresholding image',thresh,'gray')

ret, markers = cv2.connectedComponents(thresh)

marker_area = [np.sum(markers==m) for m in range(np.max(markers)) if m!=0]
largest_component = np.argmax(marker_area)+1 #Add 1 since we dropped zero above
brain_mask = markers==largest_component

brain_out = img.copy()
brain_out[brain_mask==False] = (0,0,0)

img = cv2.imread("jpg/final_fusion.jpg")
# noise removal
kernel = np.ones((3,3),np.uint8)
opening = cv2.morphologyEx(thresh,cv2.MORPH_OPEN,kernel, iterations = 2)

# sure background area
sure_bg = cv2.dilate(opening,kernel,iterations=3)

# Finding sure foreground area
dist_transform = cv2.distanceTransform(opening,cv2.DIST_L2,5)
ret, sure_fg = cv2.threshold(dist_transform,0.7*dist_transform.max(),255,0)

# Finding unknown region
sure_fg = np.uint8(sure_fg)
unknown = cv2.subtract(sure_bg,sure_fg)

# Marker labelling
ret, markers = cv2.connectedComponents(sure_fg)

# Add one to all labels so that sure background is not 0, but 1
markers = markers+1

# Now, mark the region of unknown with zero
markers[unknown==255] = 0
markers = cv2.watershed(img,markers)
img[markers == -1] = [255,0,0]

im1 = cv2.cvtColor(img,cv2.COLOR_HSV2RGB)
save_to_path="jpg/segmented.jpg"
cv2.imwrite(save_to_path, im1)
ShowImage('Watershed segmented image',im1,'gray')
```

## 5.5 Segmented Images of Various Patients:

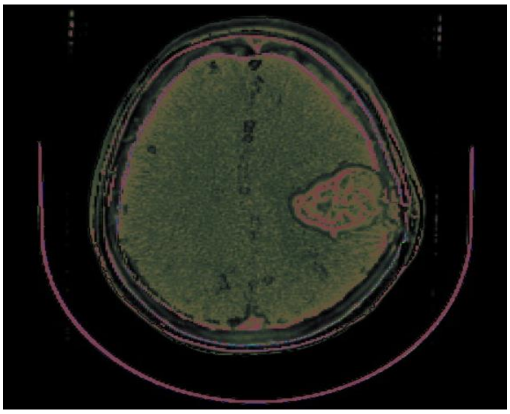

*Figure 5.1 Segmented Image of a person with brain tumor*

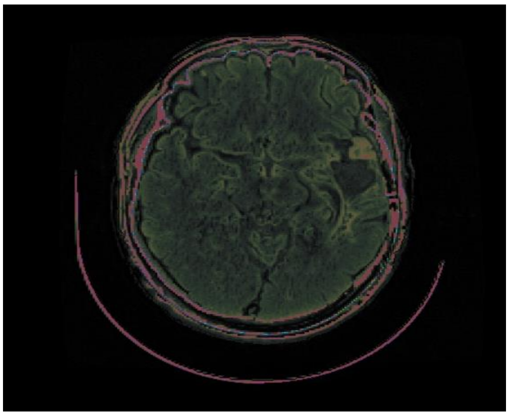

*Figure 5.2 Segmented Image of a person without any abnormalities*

# CHAPTER – 06

# GUI for Multimodal Medical Image Fusion and Analysis

Graphical User Interface (GUI) for this project is implemented. It will perform all the three tasks- Image Registration, Image Fusion, Image Segmentation on the input images and gives the final segmented image as output.

**6.1 Imports and Libraries:** A variety of essential libraries and modules are used in the script:

1. Tkinter - serves as the backbone for building the GUI.
2. PIL (Python Imaging Library)- is used for handling various image operations.
3. The filedialog module enables users to select files seamlessly.
4. The os module is utilized for performing operating system tasks.
5. numpy - is used for numerical computations.
6. matplotlib- to generates insightful plots.
7. OpenCV (cv2) - is integrated for robust image processing functionalities.
8. imageio - is employed for reading and writing image data.
9. scipy - performs scientific computing tasks.
10. argparse - is utilized for parsing command-line arguments.
11. torch- is incorporated for deep learning functionalities.
12. pywt- executes wavelet transforms efficiently.

## 6.2 GUI Implementation:

The following are the steps for implementing GUI:

- **Image Registration:** A Procrustes function is being defined which is crucial for landmark-based image registration. Leveraging Procrustes analysis, the program can align MRI and CT images accurately based on user-defined landmark points.
- **VGG19 CNN For Fusion:** The VGG19 CNN For Fusion class is implemented to encapsulate the VGG19 convolutional neural network (CNN), tailored specifically for merging information from MRI and CT scans effectively, utilizing deep learning techniques.

- **Segmentation Function:** A segmentation function is integrated into the script, leveraging OpenCV's watershed algorithm to delineate brain tumor regions within the fused image accurately. This step enables precise localization and analysis of tumor areas, enhancing clinical assessment and treatment planning.
- **Image Fusion Function:** Image fusion functionality is incorporated, utilizing wavelet transforms and the VGG19 CNN-based fusion algorithm to merge MRI and CT images effectively. By enhancing spatial and spectral features, this process facilitates more accurate brain tumor detection.

- **Registration Function:** The registration function aligns MRI and CT images by identifying corresponding landmark points provided by the user. Through Procrustes analysis and transformation matrices, this process ensures spatial consistency between modalities, enhancing fusion and analysis reliability.

- **GUI Creation:** The GUI interface includes buttons for selecting MRI and CT images, displaying the uploaded images, capturing landmark points through mouse clicks, and triggering the registration and fusion processes. This intuitive interface empowers medical professionals and researchers to interactively analyze multimodal medical images for brain tumor detection.

- **Event Handling:** Event handlers are defined within the script to capture mouse clicks on the MRI and CT image displays. These handlers facilitate the selection of landmark points required for the registration process, ensuring precise alignment between the two modalities.

- **Execution:** The GUI application runs within a main event loop, allowing continuous interaction until the user closes the window. This loop facilitates seamless image upload, registration, fusion, and result visualization, enhancing usability and workflow efficiency.

## 6.3 Graphical User Interface:

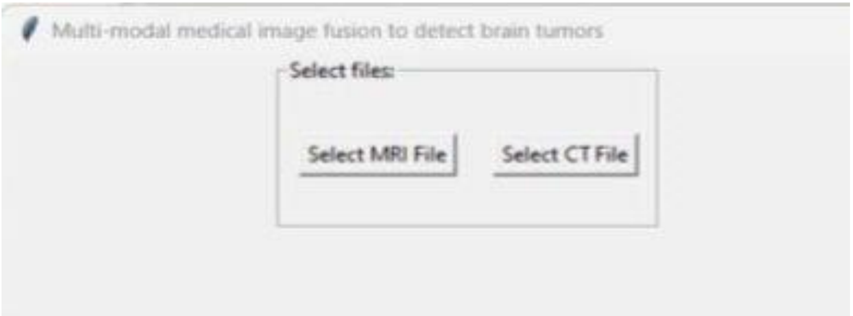


*Figure 6.1 Dialogue box to select MRI & CT Images*

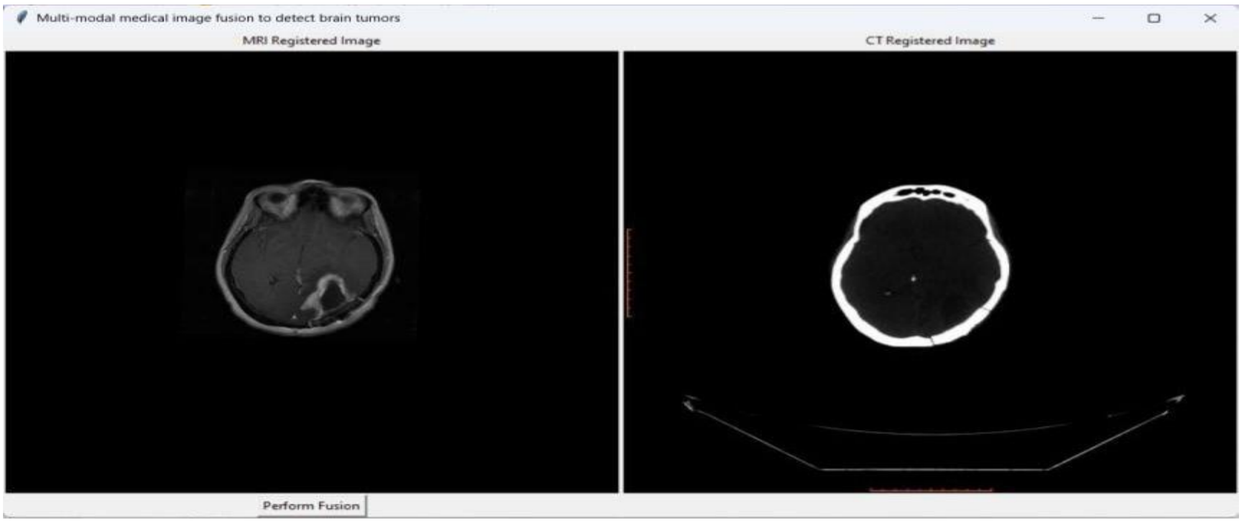


*Figure 6.2 GUI displayed MRI and CT Registered Images*

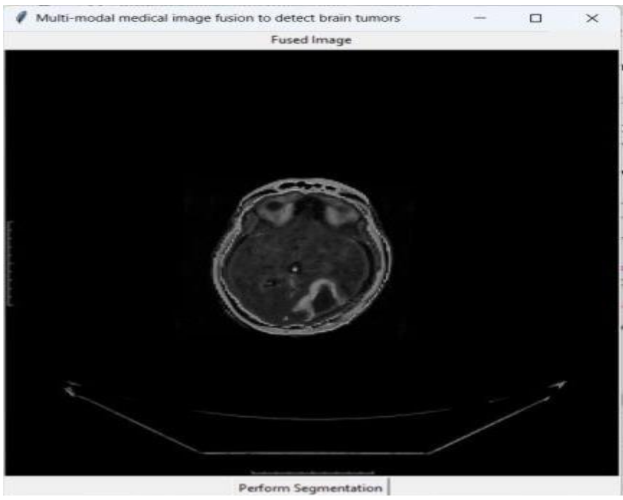


*Figure 6.3 GUI displayed Fused Image*

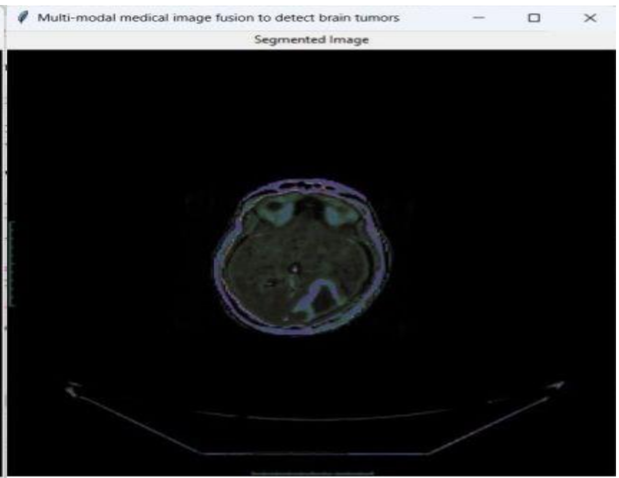


*Figure 6.4 GUI displayed Segmented Image*

# CHAPTER – 07

# RESULTS AND DISSCUSION:

## 7.1 MRI (Magnetic Resonance Imaging) Scan Images:

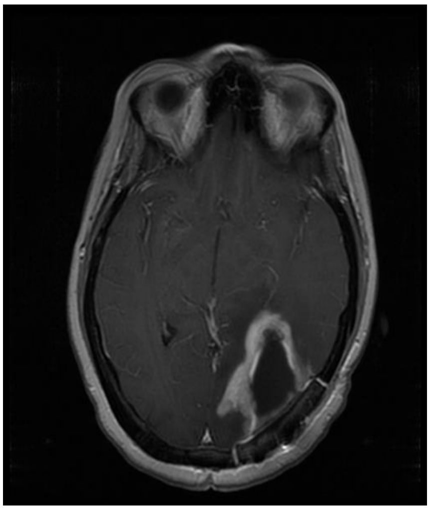

*Figure 7.1 MRI Image of Patient-1*

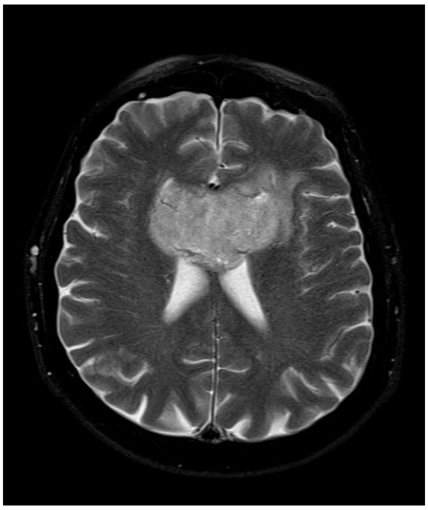

*Figure 7.2 MRI Image of Patient-2*

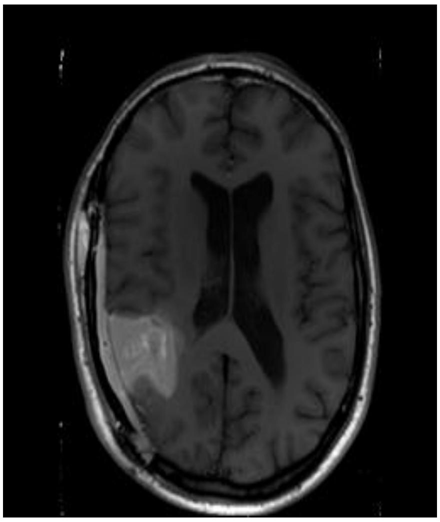

*Figure 7.3 MRI Image of Patient-3*

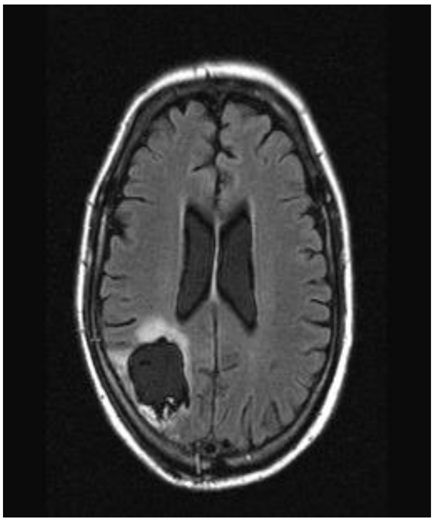

*Figure 7.4 MRI Image of Patient-4*

## 7.2 CT (Computed Tomography) Scan Images:

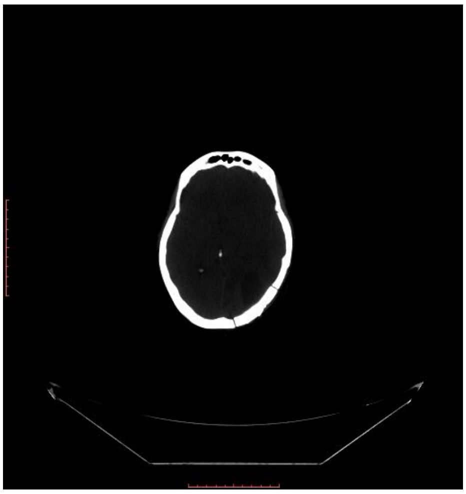

*Figure 7.5 CT Image of Patient-1*

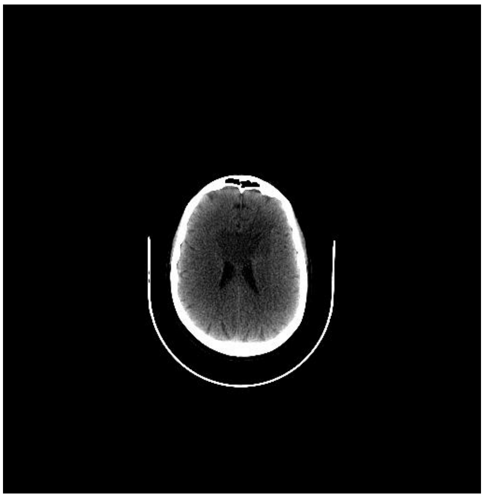

*Figure 7.6 CT Image of Patient-2*

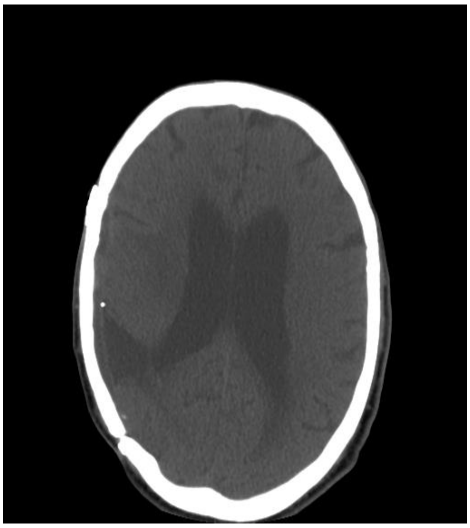

*Figure 7.7 CT Image of Patient-3*

*Figure 7.8 CT Image of Patient-4*

Above are the MRI and CT Images of four random patients.

### 7.3 Registered MRI Images of the Patients:

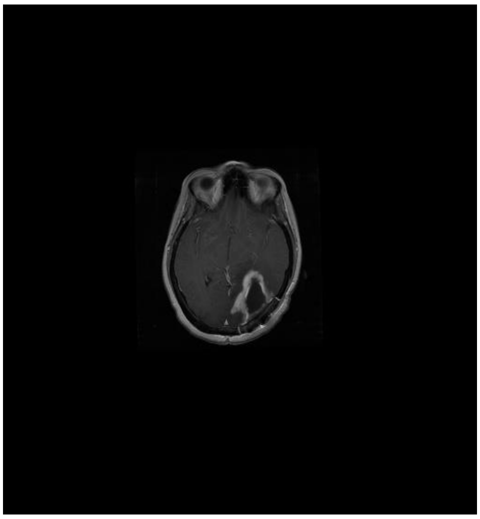

*Figure 7.9 Registered MRI Image of Patient-1*

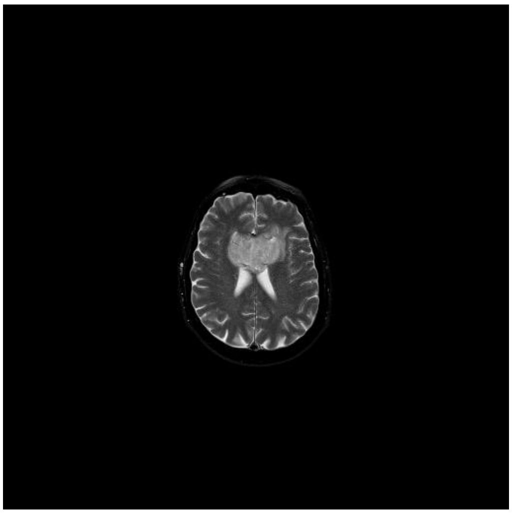

*Figure 7.10 Registered MRI Image of Patient-2*

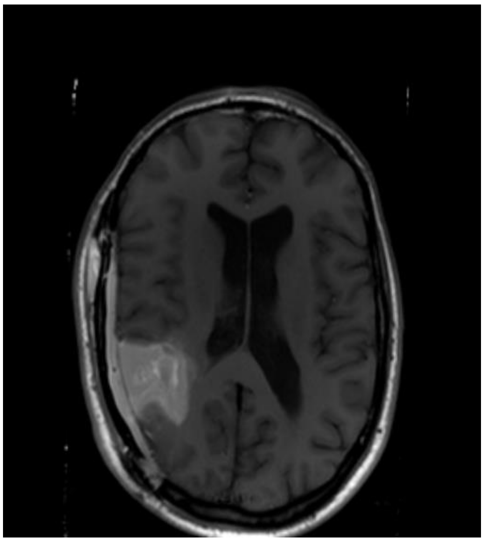

*Figure 7.11 Registered MRI Image of Patient-3*

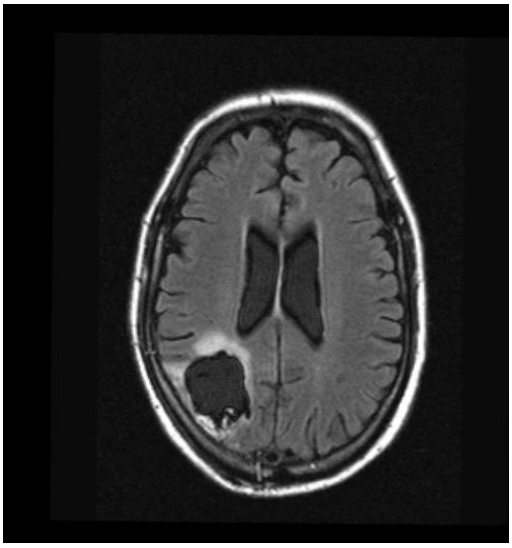

*Figure 7.12 Registered MRI Image of Patient-4*

## 7.4 Fused Images of the Patients:

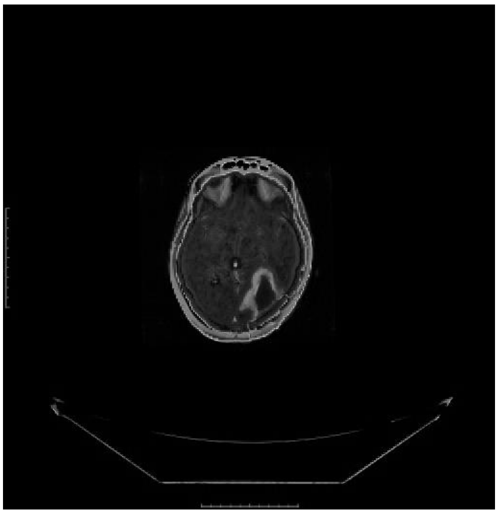

*Figure 7.13 Fused Image of Patient-1*

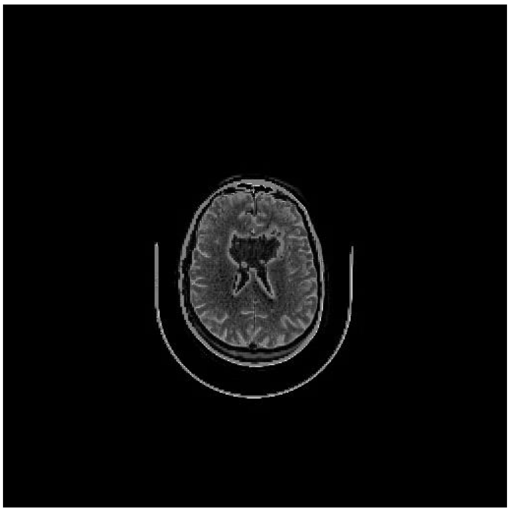

*Figure 7.14 Fused Image of Patient-2*

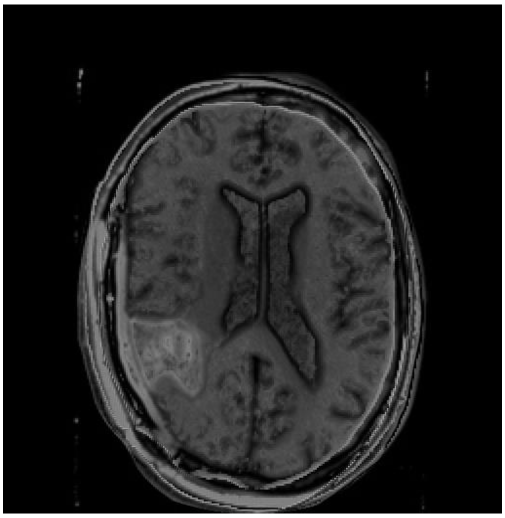

*Figure 7.15 Fused Image of Patient-3*

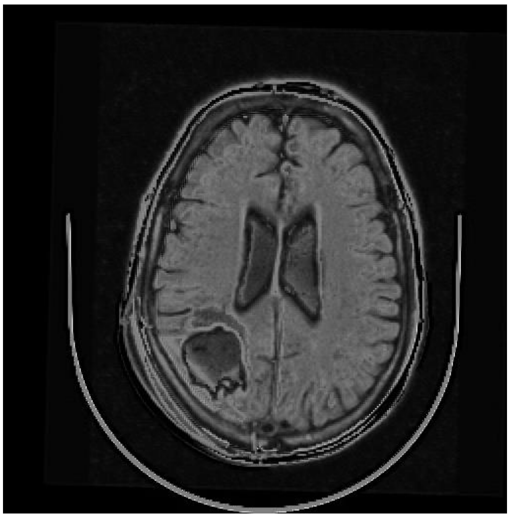

*Figure 7.16 Fused Image of Patient-4*

## 7.5 Segmented Images of the Patients:

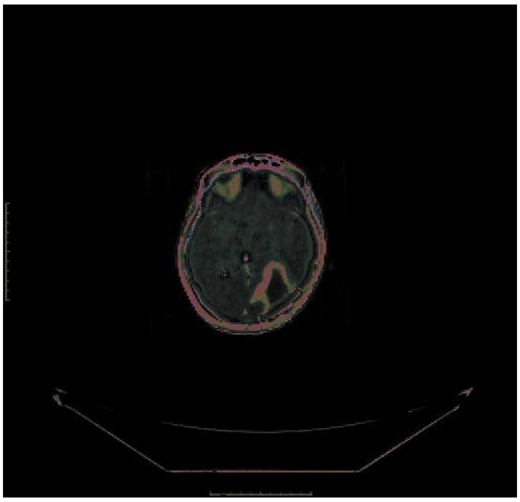

*Figure 7.17 Segmented Image of Patient-1*

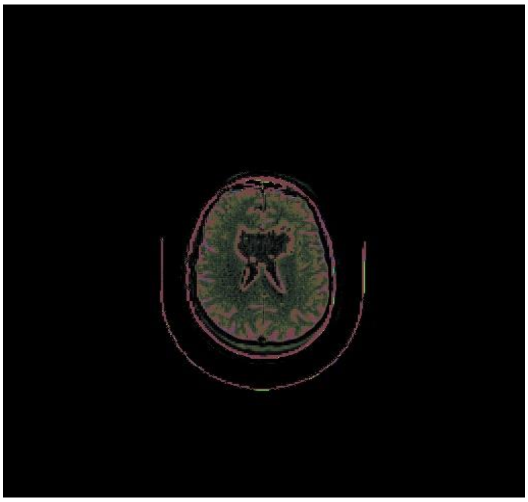

*Figure 7.18 Segmented Image of Patient-2*

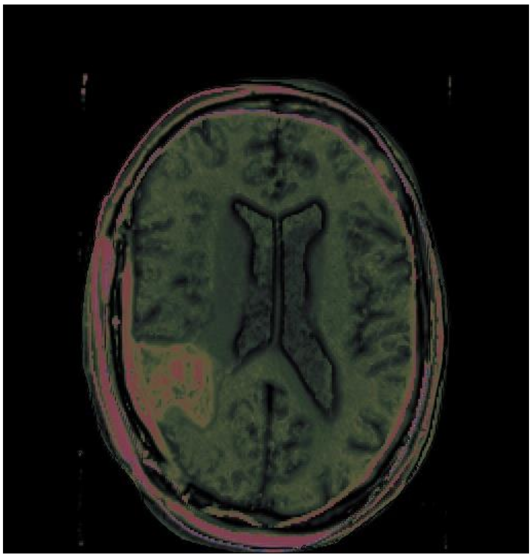

*Figure 7.19 Segmented Image of Patient-3*

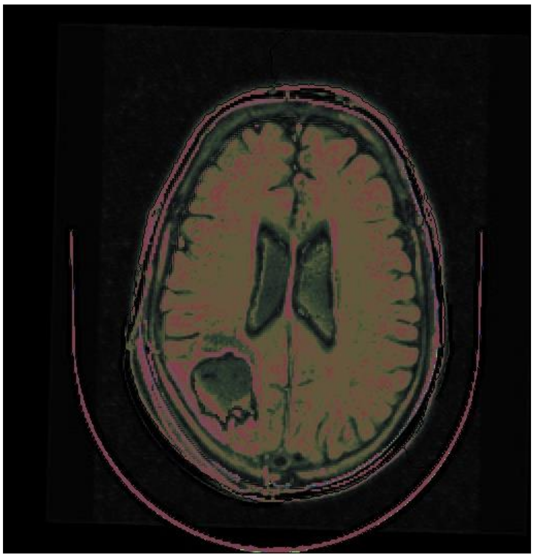

*Figure 7.20 Segmented Image of Patient-4*

In multimodal medical image fusion for brain tumor detection, user interaction varies. Around 10 to 20 clicks are needed for landmark-based registration, while fusion with the VGG-19, image segmentation with the watershed algorithm network is automated. Advanced algorithms streamline the process, improving efficiency and accuracy.

# CHAPTER - 8

# Conclusions and Future Scope

## 8.1 Comparison of VGG-19 with Other Models & Techniques:

| Aspect | VGG-19 + Watershed Model | Other Models Used for Brain Tumor Detection |
|---|---|---|
| **Feature Extraction** | Leverages pretrained VGG-19 for robust spatial and spectral features | Often require extensive retraining and tuning for specific datasets |
| **Image Fusion Quality** | Combines MRI and CT images effectively for comprehensive visualization | May focus on single modality or less effective fusion techniques |
| **Registration Accuracy** | Utilizes landmark-based registration for precise spatial alignment | May use simpler or less accurate registration methods |
| **Segmentation Precision** | Watershed algorithm provides accurate and reliable tumor boundary delineation | May use basic thresholding or less sophisticated segmentation algorithms |
| **Transfer Learning** | Benefits from transfer learning, reducing need for large training datasets | May require training from scratch, needing more data and computational power |
| **Visualization Enhancement** | Fused images enhance visibility of subtle abnormalities | May not achieve the same level of detail and clarity |
| **Ease of Integration** | GUI enables seamless integration and use in clinical settings | May lack user-friendly interfaces, hindering clinical adoption |
| **Overall Diagnostic Tool** | Provides a comprehensive tool with detailed and fused representations | Often single-modality focused, providing less comprehensive information |

*Table 8.1 Comparison of VGG-19 with other Models & Techniques*

## 8.2 Conclusions:

Through the fusion of multimodal medical images and subsequent segmentation, the project aims to improve the accuracy and efficiency of brain tumor detection and diagnosis. By combining complementary information from CT and MRI scans, the fused images provide a more comprehensive representation of the underlying anatomy, potentially leading to better clinical outcomes and patient care.

- **Enhanced Diagnostic Accuracy:** Multimodal image fusion improves brain tumor detection accuracy by integrating complementary information from CT and MRI scans.
- **Improved Clinical Workflow:** Automated fusion-based segmentation streamlines image interpretation, reducing the time and effort required for manual analysis.
- **Personalized Treatment Planning:** Accurate tumor delineation informs personalized treatment strategies, enhancing patient care and outcomes.
- **Future Research Directions:** Ongoing advancements in multimodal image analysis and deep learning hold promise for further improving diagnostic accuracy and clinical decision-making.
- **Clinical Translation:** The integration of fusion and segmentation techniques into clinical practice has the potential to revolutionize neuro-oncology by providing clinicians with advanced tools for precise diagnosis and treatment planning.
- **Impact on Patient Care:** Ultimately, the application of multimodal image fusion and segmentation technologies enhances patient care by facilitating more accurate diagnoses and tailored treatment strategies for individuals with brain tumors.

## 8.3 Future Scope:

This project on multimodal medical image fusion for brain tumor detection shows great promise and addresses a critical need in healthcare. Here are some future scopes and potential areas for further development:

- **Improved Fusion Techniques:** Although we outlined a comprehensive approach using transfer learning and wavelet transforms, there's always room for improvement in fusion techniques. Exploring different fusion algorithms or incorporating other deep learning architectures could enhance the accuracy and robustness of tumor detection.
- **Enhanced Image Segmentation:** The watershed algorithm is effective, but exploring other segmentation techniques like region-based or boundary-based methods could provide more accurate delineation of tumor boundaries. Additionally, integrating machine learning models for automated marker selection could streamline the segmentation process.
- **Integration of Additional Modalities:** Consider integrating additional imaging modalities beyond CT and MRI, such as PET (Positron Emission Tomography) or fMRI (functional MRI). Combining information from multiple modalities could provide richer data for more accurate tumor detection and characterization.
- **Interactive Visualization Tools:** Developing interactive visualization tools that allow clinicians to explore fused images and segmentation results in detail, features like 3D visualization, tumor volume calculation, and interactive manipulation of image parameters could enhance usability and clinical utility.
- **Data Augmentation and Transfer Learning:** Exploring techniques for data augmentation to expand your training dataset and improve model generalization. Continuously updating and fine-tuning your deep learning models using transfer learning with larger and more diverse datasets can further enhance performance.
- **Clinical Trials and Longitudinal Studies:** Conducting clinical trials and longitudinal studies to evaluate the long-term efficacy and impact of the technology on patient outcomes, diagnostic accuracy will be essential for gaining regulatory approval and widespread adoption in clinical practice.

## 8.4 References


[1] Enhancing Brain Tumor Detection on MRI Images Using an Innovative VGG19 Model Based Approach- Abdullah ŞENER, Burhan ERGEN-Sakarya University Journal of Science 27(5) – ISSN: 1301-4048, e-ISSN: 2147-835X – October -2023. [Pg No: 4 to 9]

[2] On the Performance of Deep Transfer Learning Networks for Brain Tumor Detection Using MR Images- SAIF AHMAD and PALLAB K.CHOUDHURY, Volume-10, IEEE ACCESS.2022.3179376, June-2022. [Pg No: 5 to 12]

[3] Image Classification of Brain Tumor Based on Enhanced VGG 19 Convolutional Neural Network- Tian C, Xi Y, Ma Y, Chen C, Wu C, Ru K, Li W and Zhao M- ISSN: 2640-1037 Volume 6, June-2023. [Pg No: 2 to 7]

[4] Image Analysis for MRI-Based Brain Tumor Classification Using Deep Learning -Krisna Nuresa Qodri, Indah Soesanti, Hanung Adi Nugroho - IJITEE, Vol. 5, No. 1, March 2021. [Pg No: 2 to 8]

[5] Detection of Brain Tumor Using Digital Image Processing- Journal of Health & Medical Informatics- Sheeba Khan- ISSN: 2157-7420- Research Article, Volume 13:06, 2022.