

COMP4211 Report: Automated Classification and Instance Segmentation of Brain Tumors in MRI Scans

Juin TAN (20913887) - jtanat@connect.ust.hk

Henry WIJAYA (20895035) - hwijaya@connect.ust.hk

Department of Computer Science and Engineering, HKUST

Prof. Dit Yan YEUNG

26th November 2024

Background Information

Brain tumors represent one of the most aggressive forms of cancer, critically affecting patient survival rates. Alarmingly, nearly 70% of individuals diagnosed with malignant brain tumors do not survive, underscoring the urgent need for effective diagnostic and treatment methodologies (Roswell Park Comprehensive Cancer Center, 2022). In modern medical practice, precise imaging techniques, particularly magnetic resonance imaging (MRI), play a pivotal role in the detection and management of brain tumors.

Given the complexity and variability of brain tumors, accurate segmentation of these tumors from MRI scans is essential for diagnosis, treatment planning, and monitoring therapeutic outcomes. For example, consider a scenario where a patient's MRI scan reveals a suspicious mass. A robust segmentation method would not only identify the tumor's exact location but also characterize its size and shape, providing critical information for subsequent treatment strategies.

Our model aims to streamline the process of reviewing brain MRI scans, reducing the time and effort required from doctors. Traditionally, clinicians have had to manually inspect each scan, meticulously searching for the presence and location of tumors. This manual review can be both labor-intensive and susceptible to human error. By leveraging our advanced tumor segmentation model, the identification of tumor locations can be automated and expedited significantly. This efficiency not only conserves valuable human resources but also accelerates the overall diagnostic process for patients. With quicker tumor detection, healthcare providers can deliver diagnosis to patients in a more timely manner, potentially facilitating earlier interventions that could improve patient outcomes and save lives.

This project introduces a novel hybrid approach that combines classification and instance segmentation models to enhance the performance of brain tumor segmentation. By utilizing an extensive Brain Tumor Detection dataset, we evaluate our model's effectiveness through various assessment measures. The incorporation of both model types allows for a more comprehensive analysis of MRI scans, yielding a solution that is both robust and generalizable.

Furthermore, our method generates detailed reports for clinicians, improving the usability of the segmentation results. These reports provide actionable insights, facilitating better communication between healthcare providers and patients. The potential applications of this project in real-world clinical settings are significant, as precise and efficient brain tumor segmentation is crucial for improving patient outcomes.

Future research directions will focus on exploring alternative feature fusion techniques and incorporating additional imaging modalities. By enhancing our method's capabilities, we aim to further improve its performance and contribute to the ongoing battle against brain cancer.

Project Overview

Our project aims to develop an automated system for the detection of cancerous cells in brain MRI scans, utilizing machine learning techniques to enhance diagnostic accuracy and efficiency. The application can be formulated as a multi-faceted machine learning problem involving two primary tasks: classification and instance segmentation, each trained on distinct datasets. By training these models on separate datasets, the overall pipeline—comprising both the classification model and the instance segmentation model—will enhance robustness and enable more accurate predictions and diagnoses of brain MRI scans. This integrative approach ensures that our final model is more resilient and effective in clinical applications.

Machine Learning Tasks

The problem can be categorized into two main machine learning tasks:

1. Classification: This task involves determining the category into which a patient's condition falls based on the MRI image. The target classes include:
 - Healthy (No Tumor)
 - Glioma Tumor
 - Meningioma Tumor
 - Pituitary Tumor
2. Instance Segmentation: This task focuses on accurately delineating tumor boundaries within the MRI images. The model identifies the presence of tumors and generates segmentation masks that highlight the tumor regions, while also categorizing the type of tumor if one is detected.

Model Input and Output

1. Classification Model:
 - Input: The input to the classification model is a pre-processed brain MRI image. This image is fed into a deep learning architecture designed for image classification, such as a convolutional neural network (CNN).
 - Output: The output is a categorical prediction indicating the patient's condition: whether they are healthy or have one of the specified tumor types.
2. Instance Segmentation Model:
 - Input: Similar to the classification model, the instance segmentation model also receives a pre-processed MRI image as input. This model utilizes architectures like U-Net, which are specifically designed for detecting objects and delineating their boundaries within images.
 - Output: The output consists of:
 - Segmentation masks that visually outline the tumor regions on the MRI image.
 - A classification result that indicates whether a tumor is present and, if so, its type (glioma, meningioma, or pituitary).

Integration and Reporting

To enhance reliability and accuracy, our system will compare the predictions from both models. A comprehensive report will be generated to inform the clinicians about the prediction result, which includes:

- The patient's MRI scan annotated with the segmentation results.
- The predicted diagnosis.
- The confidence levels associated with the predictions.
- Relevant patient information.

In cases where the models yield conflicting predictions, the report will prompt the clinician to conduct further diagnosis. This dual-model approach not only strengthens diagnostic reliability but also empowers healthcare professionals by providing them with detailed insights into the rationale behind each prediction.

The final report will serve as a crucial decision-support tool for doctors, facilitating timely and informed clinical decisions that can significantly impact patient outcomes. The sample generated reports are at the appendix of this report.

Computing Resources

The code is executed on Google Colab Pro, which provides access to the A100 GPU as the default hardware accelerator. The libraries and their corresponding versions are displayed in Table 1.

Library	Version
Keras	2.17.1
TensorFlow	2.17.1
NumPy	1.26.4
Pandas	2.2.2
Matplotlib	3.8.0
Seaborn	0.13.2
Scikit-learn	1.5.2
OpenCV (cv2)	4.10.0.84
Pillow	11.0.0
Scikit-image	0.24.0
Python	3.10.12

Table 1: Libraries and the corresponding version

Description of the Datasets

The Brain Tumor MRI Dataset utilized for training the classification model is a comprehensive collection designed for the automated classification of brain tumors in magnetic resonance imaging (MRI) scans. This dataset comprises a total of 7,023 images, categorized into four distinct classes: Glioma Tumor, Meningioma Tumor, Pituitary Tumor, and No Tumor. Notably, the "No Tumor" class images are sourced from the Br35H dataset, ensuring a diverse representation of conditions that can be identified through MRI scans.

The dataset features MRI images of varying sizes, reflecting the diverse nature of clinical imaging practices. It is structured as an aggregation of three individual datasets: Figshare, SARTAJ Dataset, and Br35H. It is important to highlight that the SARTAJ dataset has been previously identified in research as having issues with the accurate categorization of Glioma Tumor class images. To enhance the integrity of the dataset, erroneous images from the SARTAJ dataset have been removed, and corresponding images from the Figshare site have been incorporated.

The dataset is organized into two subsets for model training and evaluation, with the training set containing 5,712 images and the testing set comprising 1,311 images. This composition is summarized in Table 2, which outlines the number of samples per category. Additionally, the distribution of samples across the different categories is visually represented in Figure 1, which displays a histogram illustrating the number of images for each class.

Training Dataset	Number of Samples	Testing Dataset	Number of Samples
Glioma Tumor	1331	Glioma Tumor	300
Pituitary Tumor	1457	Pituitary Tumor	300
Meningioma Tumor	1409	Meningioma Tumor	306
Healthy (No Tumor)	1595	Healthy (No Tumor)	405

Table 2: Description of the Classification Dataset

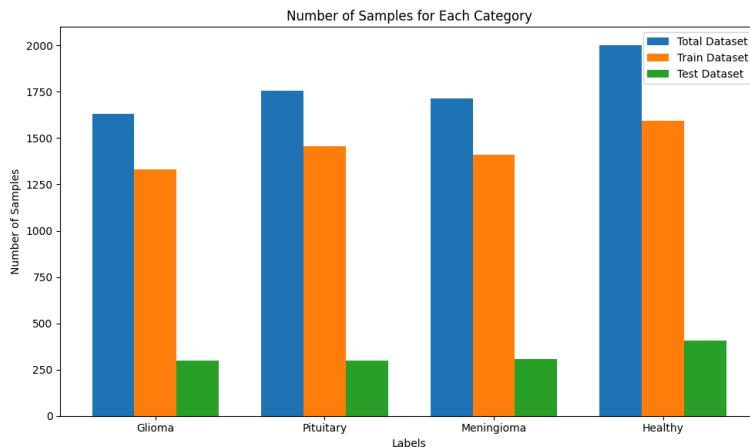


Figure 1: Distribution of the categories of the Classification Dataset

The Brain Tumor Detection dataset utilized for training the instance segmentation model is a comprehensive collection specifically designed for the automated detection and segmentation of brain tumors in magnetic resonance imaging (MRI) scans. This dataset comprises a total of 4,462 images, each meticulously annotated to facilitate effective machine learning training and evaluation.

In terms of image specifications, each image in the dataset has been resized to 640x640 pixels using a stretching method to ensure uniformity. The dataset employs auto-orientation of pixel data, with EXIF-orientation stripping applied to standardize the orientation for analysis, ensuring consistency across all images.

Annotations are provided in the COCO segmentation format. The dataset is classified into four main categories: Glioma Tumor, Meningioma Tumor, Pituitary Tumor, and No Tumor. Notably, all classes except "No Tumor" are labeled with segmentation lines, while "No Tumor" is represented by a bounding box that encompasses the entire brain. Although the "No Tumor" class is somewhat underrepresented in the dataset, it is believed that this will not adversely affect the model's performance, as it is a distinct category with unique characteristics.

The dataset is divided into three subsets for model training and evaluation: the training set contains 3,127 samples, the validation set comprises 891 samples, and the testing set includes 444 samples. This structured composition enhances the dataset's utility for developing robust instance segmentation models in clinical applications.

Training Dataset	Number of Samples	Testing Dataset	Number of Samples	Validation Dataset	Number of Samples
Glioma Tumor	943	Glioma Tumor	133	Glioma Tumor	268
Pituitary Tumor	834	Pituitary Tumor	121	Pituitary Tumor	237
Meningioma Tumor	859	Meningioma Tumor	120	Meningioma Tumor	244
Healthy (No Tumor)	491	Healthy (No Tumor)	70	Healthy (No Tumor)	142

Table 3: Description of the Segmentation Dataset

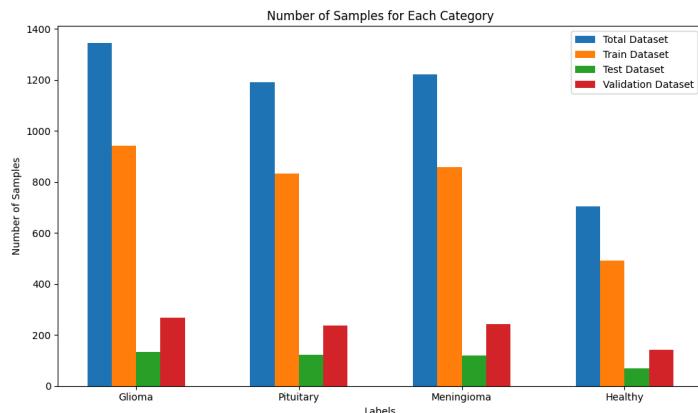


Figure 2: Distribution of the categories of the Segmentation Dataset

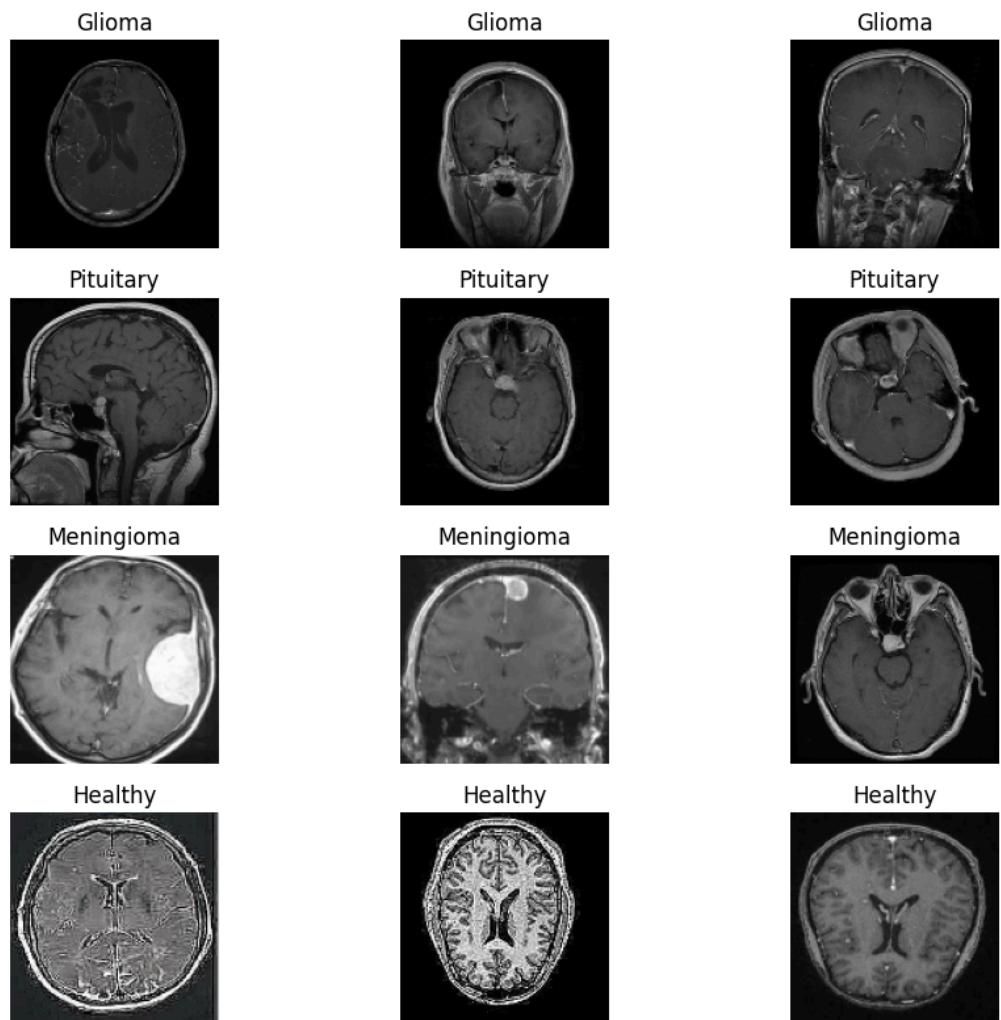


Figure 3: Visualization of the Classification Dataset

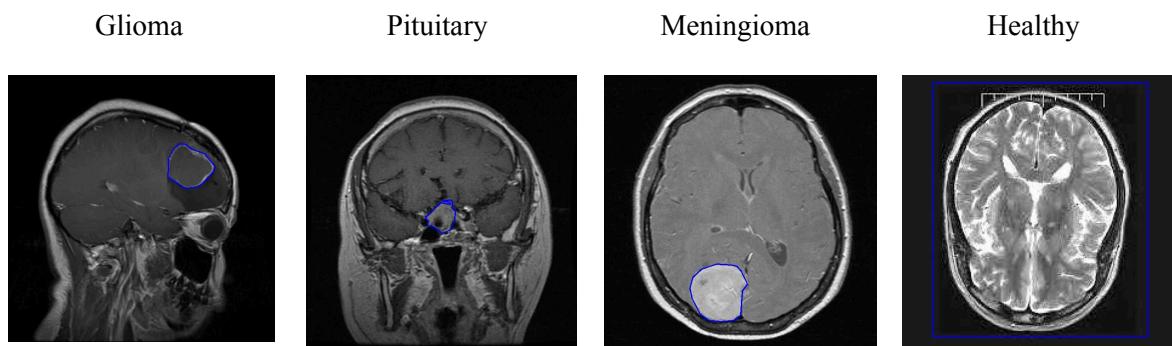


Figure 4: Visualization of the Segmentation Dataset

Machine Learning Tasks

Machine Learning Task 1: Brain Tumor Classification Model

Preprocessing Techniques

All the images in the classification dataset are resized to a dimension of (224, 224) and the labels are encoded using one-hot encoding for training. Three different preprocessing techniques are tested against the same CNN models architecture to analyze if there are significant differences in performance between different preprocessing techniques for the Brain Tumor MRI Scan classification task.

Preprocessing technique 1: Noise Reduction

Noise in input images can significantly impact the performance of image classification models. To mitigate this, noise reduction preprocessing techniques are employed. The noise reduction technique leverages the Gaussian Blur algorithm, which applies a Gaussian kernel to the input image, smoothing it and reducing high-frequency noise. The function uses a kernel size of (5, 5) and a standard deviation of 0, providing an optimal balance between noise reduction and feature preservation. By incorporating this noise reduction preprocessing, the input data presented to the classification model exhibits an enhanced signal-to-noise ratio, potentially improving model performance. Sample preprocessed images are shown in *Figure 5*.

Preprocessing technique 2: Contrast Enhancement

Suboptimal image contrast can obscure distinctive features, hindering effective classification. To address this, the contrast enhancement preprocessing technique is employed. The technique leverages Contrast Limited Adaptive Histogram Equalization (CLAHE), which divides the image into contextual regions and applies histogram equalization independently on each region. This enhances local contrast while preventing noise amplification. The implementation uses a clip limit of 2.0 and a tile grid size of (8, 8), empirically determined to provide a balanced improvement in contrast without excessive noise. Sample preprocessed images are shown in *Figure 5*.

Preprocessing technique 3: Stacked Edges

In the context of MRI imaging, brain tumor regions often exhibit a fuzzy appearance, which may hinder the ability of Convolutional Neural Networks (CNNs) to effectively learn and identify the tumor features. To mitigate this challenge, edge detection and stacked edges serve as a critical preprocessing step in image classification, as it emphasizes the salient features within the images.

This technique employs two primary algorithms:

1. Canny Edge Detection: A multi-stage process that detects strong edges while reducing noise. It outputs a binary image marking the most prominent edges.
2. Sobel Filter: Computes the gradient magnitude to identify edges in both horizontal and vertical directions, emphasizing areas of high spatial frequency.

The results from Canny and Sobel are stacked with the original grayscale image, creating a multi-channel representation that retains essential features while enhancing edge information. Sample preprocessed images are shown in *Figure 5*.

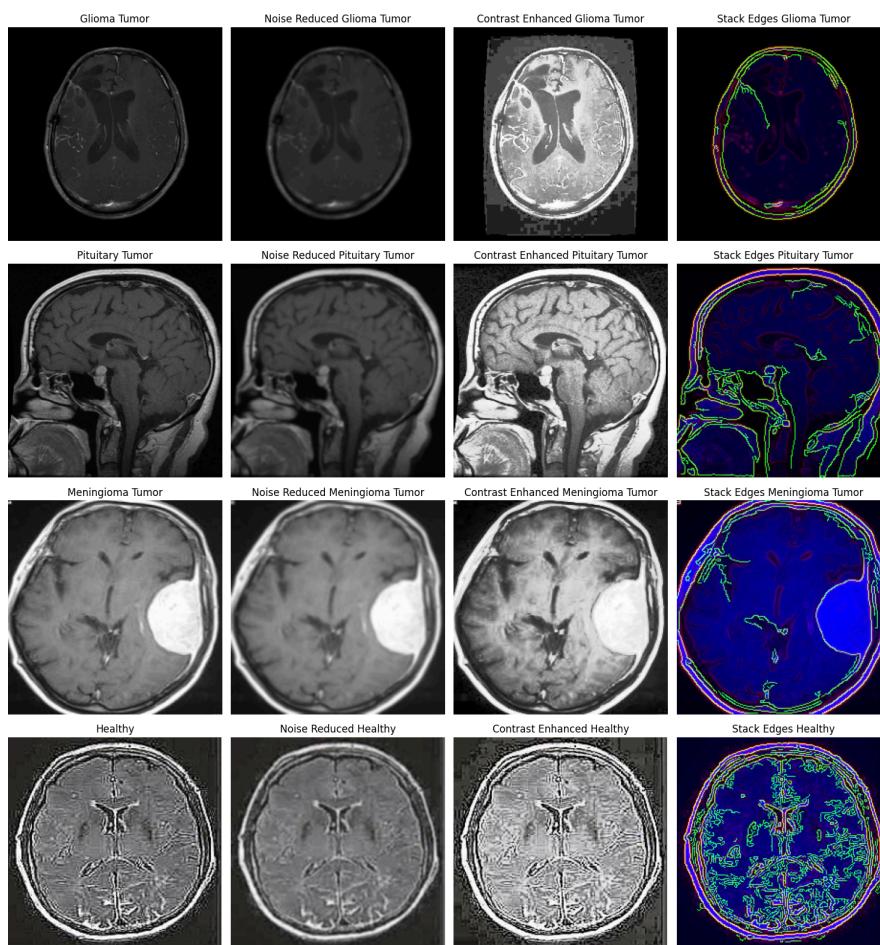


Figure 5: Original image and the preprocessed images

Machine Learning Models

We proposed three distinct machine learning models for the classification of brain tumors: the traditional Convolutional Neural Network (CNN) model, the ResNet50 + CNN model, and the Vision Transformer (ViT) model. The ResNet50 + CNN model is categorized as a CNN, whereas the ViT model is based on transformer architecture.

Table 4 provides a summary of the models utilized for the brain tumor classification task, along with their respective preprocessing techniques. Notably, the CNN model was trained using all available preprocessing techniques to evaluate which methods would yield the optimal results for this classification task. In contrast, the ResNet50 and ViT models were employed to assess their performance relative to the CNN model and to investigate how their results compare in the context of brain tumor classification.

Model	Preprocessing Technique Used
CNN model	Resize images to (224, 224)
CNN model	Resize images to (224, 224) + Noise Reduction
CNN model	Resize images to (224, 224) + Contrast Enhancement
CNN model	Resize images to (224, 224) + Stacked Edges
ResNet50 + CNN	Resize images to (224, 224)
ResNet50 + CNN	Resize images to (224, 224) + Stacked Edges
Vision Transformers (ViT)	Resize images to (224, 224)
Vision Transformers (ViT)	Resize images to (224, 224) + Stacked Edges

Table 4: List of models and the preprocessing technique used

CNN Model

A Convolutional Neural Network (CNN) model was developed and it processes input images to predict their class labels using a series of convolutional, pooling, and fully connected layers. Three distinct preprocessing pipelines and the original data were systematically applied to the model to determine their respective impacts on performance metrics. The datasets that yield the highest performance metric will be utilized for all subsequent models following the CNN model. This approach ensures that the most effective preprocessing method is consistently applied in the evaluation of later models. The parameter settings for the CNN model are shown in Table 5. The parameter settings were set based on different experiments to figure out the parameters that yield the best results.

Parameters	Settings
Epochs	50
Optimizer	Adam
Regularization	L2 Regularization (0.0001)
Learning Rate	0.0005
Batch Size	64
Loss	Categorical Cross Entropy Loss
Metric	Accuracy

Table 5: Parameter settings for the CNN model

Model Architecture

1. Input Layer:
 - The model accepts input images of shape (IMG_HEIGHT, IMG_WIDTH, 3), where 3 represents the RGB color channels.
2. Convolutional Layers:
 - First Convolutional Layer:
 - 32 filters of size (3, 3) with ReLU activation. This layer extracts basic features from the input images.
 - Second Convolutional Layer:
 - 64 filters of size (3, 3) with ReLU activation. This layer captures more complex patterns.
 - Third Convolutional Layer:
 - 128 filters of size (3, 3) with ReLU activation. This further deepens the feature extraction.
 - Fourth Convolutional Layer:
 - Another 128 filters of size (3, 3) with ReLU activation, enhancing the model's ability to learn intricate features.
3. Pooling Layers:
 - After the second and fourth convolutional layers, MaxPooling2D layers with a pool size of (2, 2) are used to downsample the feature maps, reducing spatial dimensions and retaining essential features.
4. Dropout Layers:
 - Dropout is applied after the first and second pooling layers, with rates of 0.3 and 0.4, respectively, to prevent overfitting by randomly setting a fraction of input units to zero during training.
 - Additional Dropout layers with a rate of 0.5 are included after the dense layers to further reduce overfitting.
5. Flatten Layer:
 - The output from the last pooling layer is flattened to convert the 2D feature maps into a 1D feature vector, making it suitable for fully connected layers.

6. Fully Connected Layers:

- First Dense Layer:
 - Contains 64 units with ReLU activation and L2 regularization ($l2(0.0001)$) to penalize large weights, promoting model generalization.
- Second Dense Layer:
 - Contains 32 units with ReLU activation and L2 regularization, continuing the process of feature learning.

7. Output Layer:

- The final dense layer has `num_classes` units (representing the number of classes in the classification task) with a softmax activation function, which outputs a probability distribution over the classes.

ResNet50 + CNN Model

This is a Convolutional Neural Network (CNN) model and it processes input images to predict their class labels through a series of convolutional, pooling, and fully connected layers. This model utilizes the ResNet50 architecture as its backbone, which has been pre-trained on the ImageNet dataset. The base model is frozen to retain its learned features while custom layers are added to adapt it for the specific classification task. The architecture includes a Global Average Pooling layer followed by two dense layers with dropout regularization to mitigate overfitting. The model was trained on data that was partitioned into training and validation sets, ensuring robust evaluation. The parameter settings for the ResNet50-based model are shown in Table 6. These parameters were determined through various experiments aimed at optimizing performance metrics. The model architecture is shown at the appendix of the report.

Parameters	Settings
Epochs	50
Optimizer	Adam
Learning Rate	0.0001
Batch Size	128
Loss	Categorical Cross Entropy Loss
Metric	Accuracy

Table 6: Parameter settings for the ResNet50 + CNN model

Model Architecture

1. Input Layer:
 - The model accepts input images of shape (IMG_HEIGHT, IMG_WIDTH, 3), where 3 represents the RGB color channels.
2. Base Model:
 - The architecture employs the ResNet50 model without the top classification layers, allowing for feature extraction from the input images.
3. Pooling Layer:
 - A Global Average Pooling layer is implemented to reduce the spatial dimensions of the feature maps while retaining essential information.
4. Flatten Layer:
 - The output from the last pooling layer is flattened to convert the 2D feature maps into a 1D feature vector, making it suitable for fully connected layers.
5. Dropout Layers:
 - Dropout is applied after the dense layers to prevent overfitting, with a rate of 0.2 for the first dense layer.
6. Fully Connected Layers:
 - First Dense Layer: Contains 256 units with ReLU activation, promoting non-linearity in feature representation.
 - Second Dense Layer: Contains 128 units with ReLU activation, further refining the model's ability to learn complex patterns.
7. Output Layer:
 - The final dense layer comprises num_classes units, utilizing a softmax activation function to produce a probability distribution over the class labels.

Vision Transformers (ViT) Model

This Vision Transformer (ViT) model utilizes a transformer-based architecture to process input images and predict their class labels. The ViT model operates by dividing input images into smaller patches, allowing the model to capture spatial relationships among the patches. The architecture consists of several key components: a dense layer for initial feature extraction, multiple multi-head attention layers for capturing dependencies among patches, and a global average pooling layer to condense the feature representation before classification. The parameter settings for the Vision Transformer model are shown in Table 7. These parameters were carefully chosen based on experimental trials aimed at optimizing performance metrics.

Parameters	Settings
Epochs	50
Optimizer	Adam
Learning Rate	0.0001
Batch Size	32
Loss	Categorical Cross Entropy Loss
Metric	Accuracy

Table 7: Parameter settings for the Vision Transformer (ViT) model

Model Architecture

1. Input Layer:
 - The model accepts input images of shape (IMG_HEIGHT, IMG_WIDTH, 3), where 3 represents the RGB color channels.
2. Patch Extraction:
 - Input images are divided into patches of size (16, 16) using the `tf.image.extract_patches` function. This process converts each image into a sequence of smaller patches, reshaping the input to (batch_size, num_patches, patch_size * patch_size * channels).
3. Dense Layer:
 - A dense layer with $d_{model} = 128$ units and ReLU activation is applied to the flattened patches, facilitating initial feature extraction from the input data.
4. Multi-Head Attention Layers:
 - The model incorporates four multi-head attention layers, each with 8 heads. These layers enable the model to capture complex dependencies among the patches, enhancing its ability to learn contextual relationships.
5. Global Average Pooling Layer:
 - A `GlobalAveragePooling1D` layer is utilized to reduce the output of the last attention layer to a fixed-size representation. This layer condenses the information across all patches, resulting in a output shape of (batch_size, d_{model}).
6. Output Layer:
 - The final dense layer comprises $num_classes$ units, employing a softmax activation function to produce a probability distribution over the class labels.

Results Obtained

Results Obtained on CNN model using different Preprocessing Techniques

Preprocessing Technique	Results	Prediction Label vs Actual Label																														
None	<ul style="list-style-type: none"> - Precision: 0.95 - Recall: 0.95 - F1-Score: 0.95 - Accuracy: 0.95 	<table border="1" style="margin-left: auto; margin-right: auto;"> <caption>Confusion Matrix</caption> <thead> <tr> <th colspan="2" rowspan="2"></th> <th colspan="3">Predicted Label</th> </tr> <tr> <th>Gloma</th> <th>Pituitary</th> <th>Meningioma</th> <th>Healthy</th> </tr> </thead> <tbody> <tr> <th rowspan="2">True Label</th> <th>Gloma</th> <td>271</td> <td>1</td> <td>23</td> <td>0</td> </tr> <tr> <th>Pituitary</th> <td>2</td> <td>265</td> <td>3</td> <td>1</td> </tr> <tr> <th rowspan="2">Meningioma</th> <td>8</td> <td>9</td> <td>230</td> <td>11</td> </tr> <tr> <th>Healthy</th> <td>1</td> <td>1</td> <td>3</td> <td>330</td> </tr> </tbody> </table>			Predicted Label			Gloma	Pituitary	Meningioma	Healthy	True Label	Gloma	271	1	23	0	Pituitary	2	265	3	1	Meningioma	8	9	230	11	Healthy	1	1	3	330
		Predicted Label																														
		Gloma	Pituitary	Meningioma	Healthy																											
True Label	Gloma	271	1	23	0																											
	Pituitary	2	265	3	1																											
Meningioma	8	9	230	11																												
	Healthy	1	1	3	330																											
Noise Reduction	<ul style="list-style-type: none"> - Precision: 0.92 - Recall: 0.92 - F1-Score: 0.92 - Accuracy: 0.92 	<table border="1" style="margin-left: auto; margin-right: auto;"> <caption>Confusion Matrix</caption> <thead> <tr> <th colspan="2" rowspan="2"></th> <th colspan="3">Predicted Label</th> </tr> <tr> <th>Gloma</th> <th>Pituitary</th> <th>Meningioma</th> <th>Healthy</th> </tr> </thead> <tbody> <tr> <th rowspan="2">True Label</th> <th>Gloma</th> <td>244</td> <td>3</td> <td>28</td> <td>0</td> </tr> <tr> <th>Pituitary</th> <td>1</td> <td>286</td> <td>10</td> <td>0</td> </tr> <tr> <th rowspan="2">Meningioma</th> <td>9</td> <td>18</td> <td>235</td> <td>11</td> </tr> <tr> <th>Healthy</th> <td>2</td> <td>4</td> <td>2</td> <td>306</td> </tr> </tbody> </table>			Predicted Label			Gloma	Pituitary	Meningioma	Healthy	True Label	Gloma	244	3	28	0	Pituitary	1	286	10	0	Meningioma	9	18	235	11	Healthy	2	4	2	306
		Predicted Label																														
		Gloma	Pituitary	Meningioma	Healthy																											
True Label	Gloma	244	3	28	0																											
	Pituitary	1	286	10	0																											
Meningioma	9	18	235	11																												
	Healthy	2	4	2	306																											
Contrast Enhancement	<ul style="list-style-type: none"> - Precision: 0.94 - Recall: 0.94 - F1-Score: 0.94 - Accuracy: 0.94 	<table border="1" style="margin-left: auto; margin-right: auto;"> <caption>Confusion Matrix</caption> <thead> <tr> <th colspan="2" rowspan="2"></th> <th colspan="3">Predicted Label</th> </tr> <tr> <th>Gloma</th> <th>Pituitary</th> <th>Meningioma</th> <th>Healthy</th> </tr> </thead> <tbody> <tr> <th rowspan="2">True Label</th> <th>Gloma</th> <td>271</td> <td>0</td> <td>22</td> <td>0</td> </tr> <tr> <th>Pituitary</th> <td>3</td> <td>286</td> <td>7</td> <td>0</td> </tr> <tr> <th rowspan="2">Meningioma</th> <td>8</td> <td>8</td> <td>226</td> <td>9</td> </tr> <tr> <th>Healthy</th> <td>1</td> <td>4</td> <td>6</td> <td>308</td> </tr> </tbody> </table>			Predicted Label			Gloma	Pituitary	Meningioma	Healthy	True Label	Gloma	271	0	22	0	Pituitary	3	286	7	0	Meningioma	8	8	226	9	Healthy	1	4	6	308
		Predicted Label																														
		Gloma	Pituitary	Meningioma	Healthy																											
True Label	Gloma	271	0	22	0																											
	Pituitary	3	286	7	0																											
Meningioma	8	8	226	9																												
	Healthy	1	4	6	308																											
Stacked Edges	<ul style="list-style-type: none"> - Precision: 0.94 - Recall: 0.94 - F1-Score: 0.94 - Accuracy: 0.94 	<table border="1" style="margin-left: auto; margin-right: auto;"> <caption>Confusion Matrix</caption> <thead> <tr> <th colspan="2" rowspan="2"></th> <th colspan="3">Predicted Label</th> </tr> <tr> <th>Gloma</th> <th>Pituitary</th> <th>Meningioma</th> <th>Healthy</th> </tr> </thead> <tbody> <tr> <th rowspan="2">True Label</th> <th>Gloma</th> <td>234</td> <td>3</td> <td>25</td> <td>0</td> </tr> <tr> <th>Pituitary</th> <td>0</td> <td>283</td> <td>12</td> <td>2</td> </tr> <tr> <th rowspan="2">Meningioma</th> <td>4</td> <td>11</td> <td>244</td> <td>8</td> </tr> <tr> <th>Healthy</th> <td>0</td> <td>3</td> <td>7</td> <td>323</td> </tr> </tbody> </table>			Predicted Label			Gloma	Pituitary	Meningioma	Healthy	True Label	Gloma	234	3	25	0	Pituitary	0	283	12	2	Meningioma	4	11	244	8	Healthy	0	3	7	323
		Predicted Label																														
		Gloma	Pituitary	Meningioma	Healthy																											
True Label	Gloma	234	3	25	0																											
	Pituitary	0	283	12	2																											
Meningioma	4	11	244	8																												
	Healthy	0	3	7	323																											

Table 8: Results obtained on the CNN model of different preprocessed dataset

Considering all four performance metrics, the Convolutional Neural Network (CNN) model achieved the highest results when no preprocessing was applied. This was followed by the contrast enhancement technique and the stacked edge detection technique, with the noise reduction technique yielding the lowest performance metrics. The superior performance of the no-preprocessing approach can be attributed to the careful assembly of our dataset, which integrates samples from three distinct individual datasets, thereby effectively addressing potential class imbalance issues. Moreover, the dataset has been preprocessed to eliminate problematic samples that could negatively impact model performance. Given the robustness and high quality of the dataset, the application of preprocessing techniques does not significantly enhance model performance.

While both the contrast enhancement technique and the stacked edges technique were beneficial in improving the quality of the brain MRI images, neither achieved the performance levels observed with the unprocessed dataset. The findings indicate that although these techniques can enhance image quality, they do not provide a substantial advantage in terms of classification accuracy when compared to the raw data. Conversely, the noise reduction technique resulted in the most significant decline across all performance metrics. This suggests that, despite the common belief that noise reduction can enhance image quality, it may have inadvertently removed essential features or details crucial for classification, ultimately leading to decreased model performance.

It is essential to highlight that all preprocessing techniques, along with the original dataset, exhibit suboptimal performance in classifying glioma tumors, frequently misclassifying them as meningioma tumors, as illustrated in the accompanying heatmaps. This misclassification may arise from several factors inherent to the dataset and the characteristics of the tumors themselves. Gliomas and meningiomas can present similar imaging features, particularly in specific MRI sequences, which may lead to confusion during the classification process. Additionally, the heterogeneity of glioma tumors—varying in both their characteristics and anatomical locations—complicates the model's ability to accurately capture these variations. To address these challenges, potential improvements could include augmenting the dataset with more glioma samples, implementing advanced preprocessing techniques, or utilizing sophisticated model architectures that are better suited for distinguishing between these closely related tumor types.

In light of these findings, we will proceed with the no-preprocessing dataset and the stacked edges dataset for training the ResNet50 + CNN model and the Vision Transformer (ViT) model. These two datasets demonstrated the best performance, allowing for further investigation into whether preprocessing or the absence of preprocessing yields superior results in subsequent analyses. This approach will facilitate a comprehensive understanding of the impact of various preprocessing techniques on model performance.

Results Obtained on different Models

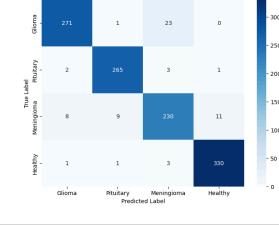
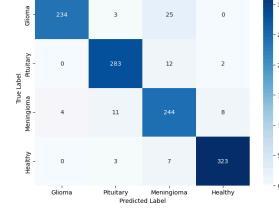
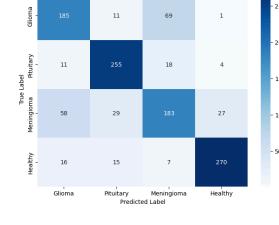
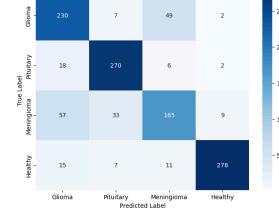
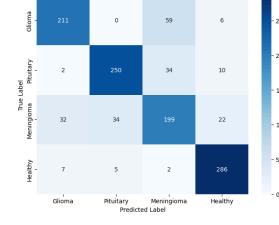
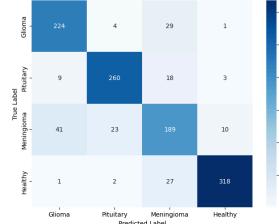
Model	Results	Prediction Label vs Actual Label																														
CNN (No preprocessing)	<ul style="list-style-type: none"> Precision: 0.95 Recall: 0.95 F1-Score: 0.95 Accuracy: 0.95 	 <table border="1"> <caption>Confusion Matrix for CNN (No preprocessing)</caption> <thead> <tr> <th colspan="2" rowspan="2">True Label</th> <th colspan="4">Predicted Label</th> </tr> <tr> <th>Gloma</th> <th>Pituitary</th> <th>Meningoma</th> <th>Healthy</th> </tr> </thead> <tbody> <tr> <th rowspan="4">Gloma</th> <td>271</td> <td>1</td> <td>23</td> <td>0</td> </tr> <tr> <th rowspan="4">Pituitary</th> <td>2</td> <td>265</td> <td>3</td> <td>1</td> </tr> <tr> <th rowspan="4">Meningoma</th> <td>8</td> <td>9</td> <td>230</td> <td>11</td> </tr> <tr> <td>1</td> <td>1</td> <td>3</td> <td>300</td> </tr> </tbody> </table>	True Label		Predicted Label				Gloma	Pituitary	Meningoma	Healthy	Gloma	271	1	23	0	Pituitary	2	265	3	1	Meningoma	8	9	230	11	1	1	3	300	
True Label		Predicted Label																														
		Gloma	Pituitary	Meningoma	Healthy																											
Gloma	271	1	23	0																												
	Pituitary	2	265	3	1																											
		Meningoma	8	9	230	11																										
			1	1	3	300																										
CNN (Stacked Edges)			<ul style="list-style-type: none"> Precision: 0.94 Recall: 0.94 F1-Score: 0.94 Accuracy: 0.94 	 <table border="1"> <caption>Confusion Matrix for CNN (Stacked Edges)</caption> <thead> <tr> <th colspan="2" rowspan="2">True Label</th> <th colspan="4">Predicted Label</th> </tr> <tr> <th>Gloma</th> <th>Pituitary</th> <th>Meningoma</th> <th>Healthy</th> </tr> </thead> <tbody> <tr> <th rowspan="4">Gloma</th> <td>234</td> <td>3</td> <td>25</td> <td>0</td> </tr> <tr> <th rowspan="4">Pituitary</th> <td>0</td> <td>283</td> <td>12</td> <td>2</td> </tr> <tr> <th rowspan="4">Meningoma</th> <td>4</td> <td>11</td> <td>244</td> <td>8</td> </tr> <tr> <td>0</td> <td>3</td> <td>7</td> <td>323</td> </tr> </tbody> </table>	True Label		Predicted Label				Gloma	Pituitary	Meningoma	Healthy	Gloma	234	3	25	0	Pituitary	0	283	12	2	Meningoma	4	11	244	8	0	3	7
True Label			Predicted Label																													
		Gloma	Pituitary	Meningoma	Healthy																											
Gloma	234	3	25	0																												
	Pituitary	0	283	12	2																											
		Meningoma	4	11	244	8																										
			0	3	7	323																										
ResNet50 + CNN (No preprocessing)			<ul style="list-style-type: none"> Precision: 0.77 Recall: 0.77 F1-Score: 0.77 Accuracy: 0.77 	 <table border="1"> <caption>Confusion Matrix for ResNet50 + CNN (No preprocessing)</caption> <thead> <tr> <th colspan="2" rowspan="2">True Label</th> <th colspan="4">Predicted Label</th> </tr> <tr> <th>Gloma</th> <th>Pituitary</th> <th>Meningoma</th> <th>Healthy</th> </tr> </thead> <tbody> <tr> <th rowspan="4">Gloma</th> <td>185</td> <td>11</td> <td>69</td> <td>1</td> </tr> <tr> <th rowspan="4">Pituitary</th> <td>11</td> <td>255</td> <td>18</td> <td>4</td> </tr> <tr> <th rowspan="4">Meningoma</th> <td>58</td> <td>29</td> <td>183</td> <td>27</td> </tr> <tr> <td>16</td> <td>15</td> <td>7</td> <td>270</td> </tr> </tbody> </table>	True Label		Predicted Label				Gloma	Pituitary	Meningoma	Healthy	Gloma	185	11	69	1	Pituitary	11	255	18	4	Meningoma	58	29	183	27	16	15	7
True Label			Predicted Label																													
		Gloma	Pituitary	Meningoma	Healthy																											
Gloma	185	11	69	1																												
	Pituitary	11	255	18	4																											
		Meningoma	58	29	183	27																										
			16	15	7	270																										
ResNet50 + CNN (Stacked Edges)			<ul style="list-style-type: none"> Precision: 0.82 Recall: 0.82 F1-Score: 0.81 Accuracy: 0.81 	 <table border="1"> <caption>Confusion Matrix for ResNet50 + CNN (Stacked Edges)</caption> <thead> <tr> <th colspan="2" rowspan="2">True Label</th> <th colspan="4">Predicted Label</th> </tr> <tr> <th>Gloma</th> <th>Pituitary</th> <th>Meningoma</th> <th>Healthy</th> </tr> </thead> <tbody> <tr> <th rowspan="4">Gloma</th> <td>230</td> <td>7</td> <td>49</td> <td>2</td> </tr> <tr> <th rowspan="4">Pituitary</th> <td>18</td> <td>270</td> <td>6</td> <td>2</td> </tr> <tr> <th rowspan="4">Meningoma</th> <td>57</td> <td>33</td> <td>165</td> <td>9</td> </tr> <tr> <td>15</td> <td>7</td> <td>11</td> <td>278</td> </tr> </tbody> </table>	True Label		Predicted Label				Gloma	Pituitary	Meningoma	Healthy	Gloma	230	7	49	2	Pituitary	18	270	6	2	Meningoma	57	33	165	9	15	7	11
True Label			Predicted Label																													
		Gloma	Pituitary	Meningoma	Healthy																											
Gloma	230	7	49	2																												
	Pituitary	18	270	6	2																											
		Meningoma	57	33	165	9																										
			15	7	11	278																										
Vision Transformer (ViT) (No preprocessing)			<ul style="list-style-type: none"> Precision: 0.82 Recall: 0.82 F1-Score: 0.82 Accuracy: 0.81 	 <table border="1"> <caption>Confusion Matrix for Vision Transformer (ViT) (No preprocessing)</caption> <thead> <tr> <th colspan="2" rowspan="2">True Label</th> <th colspan="4">Predicted Label</th> </tr> <tr> <th>Gloma</th> <th>Pituitary</th> <th>Meningoma</th> <th>Healthy</th> </tr> </thead> <tbody> <tr> <th rowspan="4">Gloma</th> <td>211</td> <td>0</td> <td>59</td> <td>6</td> </tr> <tr> <th rowspan="4">Pituitary</th> <td>2</td> <td>250</td> <td>34</td> <td>10</td> </tr> <tr> <th rowspan="4">Meningoma</th> <td>32</td> <td>34</td> <td>199</td> <td>22</td> </tr> <tr> <td>7</td> <td>5</td> <td>2</td> <td>286</td> </tr> </tbody> </table>	True Label		Predicted Label				Gloma	Pituitary	Meningoma	Healthy	Gloma	211	0	59	6	Pituitary	2	250	34	10	Meningoma	32	34	199	22	7	5	2
True Label			Predicted Label																													
		Gloma	Pituitary	Meningoma	Healthy																											
Gloma	211	0	59	6																												
	Pituitary	2	250	34	10																											
		Meningoma	32	34	199	22																										
			7	5	2	286																										
Vision Transformer (ViT) (Stacked Edges)			<ul style="list-style-type: none"> Precision: 0.86 Recall: 0.86 F1-Score: 0.86 Accuracy: 0.86 	 <table border="1"> <caption>Confusion Matrix for Vision Transformer (ViT) (Stacked Edges)</caption> <thead> <tr> <th colspan="2" rowspan="2">True Label</th> <th colspan="4">Predicted Label</th> </tr> <tr> <th>Gloma</th> <th>Pituitary</th> <th>Meningoma</th> <th>Healthy</th> </tr> </thead> <tbody> <tr> <th rowspan="4">Gloma</th> <td>224</td> <td>4</td> <td>29</td> <td>1</td> </tr> <tr> <th rowspan="4">Pituitary</th> <td>9</td> <td>260</td> <td>18</td> <td>3</td> </tr> <tr> <th rowspan="4">Meningoma</th> <td>41</td> <td>23</td> <td>189</td> <td>10</td> </tr> <tr> <td>1</td> <td>2</td> <td>27</td> <td>318</td> </tr> </tbody> </table>	True Label		Predicted Label				Gloma	Pituitary	Meningoma	Healthy	Gloma	224	4	29	1	Pituitary	9	260	18	3	Meningoma	41	23	189	10	1	2	27
True Label			Predicted Label																													
		Gloma	Pituitary	Meningoma	Healthy																											
Gloma	224	4	29	1																												
	Pituitary	9	260	18	3																											
		Meningoma	41	23	189	10																										
			1	2	27	318																										

Table 9: Results obtained on different models of different preprocessed dataset

The results indicate that the effectiveness of preprocessing techniques varies significantly across different model architectures. The Convolutional Neural Network (CNN) exhibits superior performance without any preprocessing, whereas both the ResNet50 + CNN and Vision Transformer (ViT) models demonstrate enhanced performance when utilizing the stacked edges technique. When considering all performance metrics, the CNN model achieves the highest overall performance among the models evaluated, followed by the ViT model. In contrast, the ResNet50 + CNN model exhibits the lowest performance.

Despite the advanced architectures and pre-training of both the ResNet50 and Vision Transformer (ViT) models, they exhibited worse performance compared to the Convolutional Neural Network (CNN) in this study. This discrepancy can be attributed to several factors related to the nature of the dataset and the specific characteristics of the models.

Firstly, while ResNet50 and ViT are indeed robust and have been pre-trained on large natural image datasets, this pre-training does not always translate effectively to specialized tasks such as brain tumor classification from MRI images. The features learned from natural images may not adequately capture the specific visual patterns and nuances present in medical imaging. Consequently, these models may struggle to generalize to the MRI domain without additional fine-tuning on relevant medical datasets.

Moreover, the complexity of the ResNet50 and ViT architectures, while providing powerful feature extraction capabilities, can also lead to challenges in training and optimization. These models may require a more extensive dataset to fully leverage their potential. In contrast, the simpler architecture of the CNN may have been more effective for this particular dataset, allowing it to learn the pertinent features more efficiently without the need for extensive data augmentation or complex training strategies.

Additionally, the performance of the CNN with no preprocessing indicates that the raw data provided sufficient quality for effective classification. On the other hand, the ResNet50 and ViT models, which are designed to handle higher-dimensional feature spaces, may not have benefited from the same level of direct feature extraction from the raw data. Instead, they may require preprocessing techniques that enhance specific features relevant to MRI scans, which were not as effectively captured in the initial training.

In summary, while ResNet50 and ViT are more robust and sophisticated models, their performance can be hampered by the limitations of pre-training on natural images and the complexity of their architectures. These models may require tailored fine-tuning and specialized preprocessing to achieve optimal performance in the context of brain tumor classification from MRI images. Future investigations should aim to explore these adaptations to enhance the effectiveness of these advanced models in medical imaging tasks.

Machine Learning Task 2: Brain Tumor Segmentation Model

All the images in the segmentation dataset are resized to a dimension of (128, 128) and the labels are in the forms of segmentation labels, which are then turned into image masks of the tumor.

The U-Net model is designed for image segmentation tasks, particularly in biomedical imaging. It features a contracting path to capture context and an expansive path that enables precise localization.

Model Architecture:

Input Layer:

Shape: (height, width, channels) (typically (IMG_HEIGHT, IMG_WIDTH, 3) for RGB images)

Contracting Path:

Block 1:

2 x Conv2D (128 filters, (3, 3), ReLU, padding='same',
kernel_initializer='he_normal')
MaxPooling2D (pool size (2, 2))

Block 2:

2 x Conv2D (256 filters, (3, 3), ReLU, padding='same',
kernel_initializer='he_normal')
MaxPooling2D (pool size (2, 2))

Block 3:

2 x Conv2D (512 filters, (3, 3), ReLU, padding='same',
kernel_initializer='he_normal')
MaxPooling2D (pool size (2, 2))

Block 4:

2 x Conv2D (1024 filters, (3, 3), ReLU, padding='same',
kernel_initializer='he_normal')
MaxPooling2D (pool size (2, 2))

Bottleneck:

2 x Conv2D (2048 filters, (3, 3), ReLU, padding='same',
kernel_initializer='he_normal')

Expansive Path:

Block 5:

Conv2DTranspose (1024 filters, (2, 2), strides (2, 2), padding='same')
Concatenate with Block 4 output
2 x Conv2D (1024 filters, (3, 3), ReLU, padding='same',
kernel_initializer='he_normal')

Block 6:

Conv2DTranspose (512 filters, (2, 2), strides (2, 2), padding='same')
Concatenate with Block 3 output
2 x Conv2D (512 filters, (3, 3), ReLU, padding='same',
kernel_initializer='he_normal')

Block 7:

Conv2DTranspose (256 filters, (2, 2), strides (2, 2), padding='same')
Concatenate with Block 2 output

```
2 x Conv2D (256 filters, (3, 3), ReLU, padding='same',  
kernel_initializer='he_normal')
```

Block 8:

```
Conv2DTranspose (128 filters, (2, 2), strides (2, 2), padding='same')  
Concatenate with Block 1 output
```

```
2 x Conv2D (128 filters, (3, 3), ReLU, padding='same',  
kernel_initializer='he_normal')
```

Output Layer:

```
Conv2D (NUM_CLASSES filters, (1, 1), softmax activation)
```

The model is compiled using dice loss function and adam optimizer with learning rate of 0.001.

Discussion of the Results

Machine Learning Task 1: Brain Tumor Classification Model

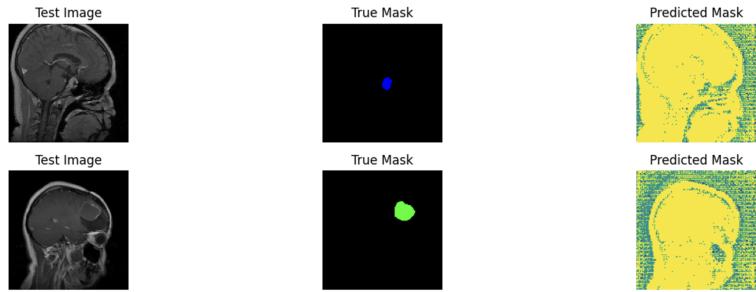
Among all the models evaluated, the Convolutional Neural Network (CNN) model without any preprocessing emerged as the best option, achieving the highest performance metrics across the board. This suggests that the CNN's architecture is well-suited for directly extracting relevant features from the robust dataset we utilized, which was carefully curated to mitigate class imbalance and eliminate problematic samples. In contrast, the Vision Transformer (ViT) model demonstrated strong performance with the stacked edges preprocessing technique, achieving notable precision, recall, F1-score, and accuracy metrics. This indicates that while the ViT architecture is capable of leveraging advanced attention mechanisms, its effectiveness is enhanced when appropriate preprocessing techniques are applied.

The choice of model and preprocessing technique involves several trade-offs. The CNN model's superior performance without preprocessing suggests ease of use and efficiency, as it does not require additional steps to enhance image quality. However, this simplicity may limit the model's ability to generalize in more complex scenarios or with less robust datasets. On the other hand, the ViT model, while benefiting from preprocessing, may introduce additional complexity in terms of computational requirements and training time, as well as the need for careful selection of preprocessing techniques that effectively enhance feature representation.

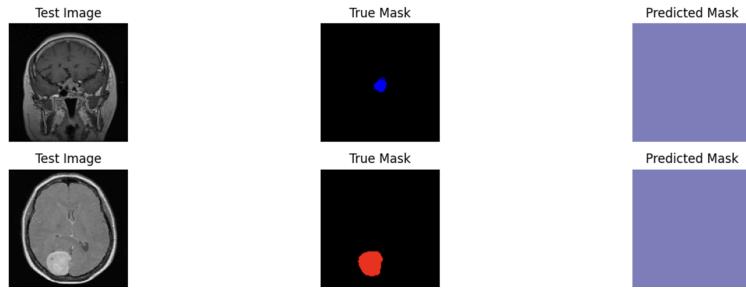
To further enhance model performance, several avenues for improvement can be explored. First, fine-tuning the ViT architecture with a focus on MRI-specific features could yield better classification results. This could involve retraining the model on our dataset with adjusted hyperparameters and additional layers to capture the nuances of brain tumor imaging. Moreover, the implementation of advanced data augmentation techniques could help increase the diversity of the training dataset, potentially improving the prediction accuracy on the glioma tumor of all models.

Machine Learning Task 2: Brain Tumor Segmentation Model

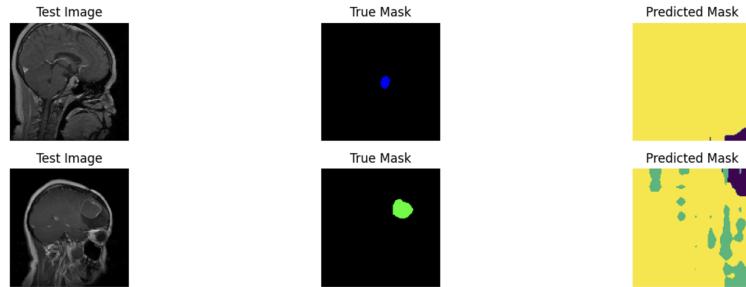
Output of the untrained U-net image segmentation model (Test Accuracy: 1.29%)



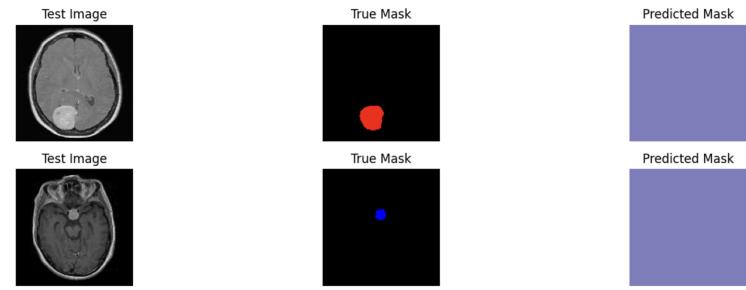
After Training U-net for 1 epochs (Test Accuracy: 0.20%)



Untrained deeplabv3+ (Test Accuracy: 3.47%)



Trained deeplabv3+ for 1 epochs (Test Accuracy: 0.20%) (dice loss)



All models struggle with training and the output converges to a single color. This issue arises because the background pixels dominate the dataset, leading the model to learn that predicting every pixel as the background minimizes the loss function. As a result, the model fails to accurately identify other classes.

Division of Labor

Juin TAN (20913887)

- Design of the pipeline
- Find and explore suitable classification and segmentation datasets
- Literature reviews
- Report (Background Information)
- Report (Project Overview)
- Report (Computing Resources)
- Report (Description of the Datasets)
- Report (Machine Learning Task 1)
- Report (Discussion of the Results Obtained for Machine Learning Task 1)
- Report (Citations)
- Report (Appendices)
- Code (Visualize segmentation from dataset [Not in submitted code])
- Code (Set up the environment)
- Code (Visualize the Distribution of the Classification and Segmentation Datasets)
- Code (Load and Visualize the Classification Dataset)
- Code (Machine Learning Task 1)
- Code (Generate Report)
- Slides
- Presentation

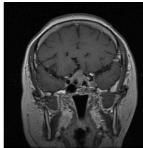
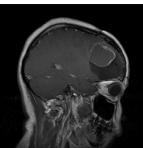
Henry WIJAYA (20895035)

- Literature reviews
- Report (Machine Learning Task 2)
- Report (Discussion of the Results Obtained for Machine Learning Task 2)
- Code (Machine Learning Task 2)

Citations

1. Roswell Park. (2022, February 16). *Survival for brain cancer*. Roswell Park Comprehensive Cancer Center. <https://www.roswellpark.org/cancer/brain/survival-rates>
2. Ullah, F., Nadeem, M., Abrar, M., Al-Razgan, M., Alfakih, T., Amin, F., & Salam, A. (2023a). Brain tumor segmentation from MRI images using handcrafted convolutional neural network. *Diagnostics*, 13(16), 2650. <https://doi.org/10.3390/diagnostics13162650>
3. Isselmou, A. E., Zhang, S., & Xu, G. (2016). A novel approach for brain tumor detection using MRI images. *Journal of Biomedical Science and Engineering*, 09(10), 44–52. <https://doi.org/10.4236/jbise.2016.910b006>
4. Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. *Lecture Notes in Computer Science*, 234–241. https://doi.org/10.1007/978-3-319-24574-4_28
5. Milletari, F., Navab, N., & Ahmadi, S.-A. (2016). V-net: Fully convolutional neural networks for volumetric medical image segmentation. *2016 Fourth International Conference on 3D Vision (3DV)*, 565–571. <https://doi.org/10.1109/3dv.2016.79>
6. Msoud Nickparvar. (2021). Brain Tumor MRI Dataset [Data set]. Kaggle. <https://doi.org/10.34740/KAGGLE/DSV/2645886>
7. Firstworkspace. (2024). BrainTumorDetection Databse. Roboflow. <https://universe.roboflow.com/firstworkspace-qsq1i/braintumordetection-agfcl>
8. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2020). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. arXiv (Cornell University). <https://doi.org/10.48550/arxiv.2010.11929>
9. He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep residual learning for image recognition. arXiv (Cornell University). <https://doi.org/10.48550/arxiv.1512.03385>

Appendices

Brain Tumor Prediction Report	Brain Tumor Prediction Report
Patient Name: CHAN Tai Man	Patient Name: CHAN Tai Man
Prediction Result: Pituitary Tumor	Prediction Result: Glioma Tumor
<p>* This prediction result has a high confidence level. However, it is always advisable to consult with a qualified medical professional for a comprehensive evaluation.</p>	<p>* The prediction results are inconsistent. It is important to consult a qualified medical professional for a thorough evaluation and diagnosis to confirm the findings.</p>
Patient's MRI Image:	Patient's MRI Image:
	
Predicted Diagnosis MRI Image:	Predicted Diagnosis MRI Image:
