BrainSight: Efficient Brain Tumor Segmentation Utilizing Computer Vision

Kent Amadeo Timotheus	2702227025
Theodore Zachary	2702244100
Albertus Edbert Chandrajaya	2702345440

Lecturer: D6705 - Dr. Nur Afny Catur Andryani, S.Si., M.Sc

School of Computer Science
Computer Science Program
BINUS UNIVERSITY
ACADEMIC YEAR 2025/2026

Abstract—Brain tumors are serious neurological conditions that require accurate and timely diagnosis to support effective treatment planning. Magnetic Resonance Imaging (MRI) is the primary modality used for brain tumor diagnosis; however, tumor identification and segmentation are still largely performed manually by medical personnel, making the process time-consuming and prone to subjectivity and inconsistency. This project proposes an automated brain tumor segmentation system using a classical machine learning approach based on Random Forest classification and patch-based feature extraction. The system processes MRI images through pre-processing, pixel-level feature extraction, model inference, and post-processing to generate binary segmentation masks that highlight tumor regions. Experiments were conducted using a public MRI dataset consisting of glioma, meningioma, and pituitary tumor images, with balanced data distribution and stratified training, validation, and testing splits. The proposed model achieved an accuracy of 95.14% and a Dice coefficient of 95.27% on unseen data, demonstrating its ability to segment tumor regions accurately and consistently. Despite challenges related to computational complexity and model size, the results indicate that the proposed system has strong potential as a supportive computer-aided diagnostic tool to improve efficiency, consistency, and reliability in MRI-based brain tumor analysis.

Keywords—Brain tumor, image segmentation, computer vision, machine learning, random forest

CHAPTER 1
INTRODUCTION

2.1 Background

Brain tumors are diseases caused by abnormal cell growth in the brain, with more than 150,000 cases occurring annually in Indonesia [1], [2]. Diagnosis using MRI still relies on manual analysis by specialists, making the process slow and inefficient. With advances in computer vision, MRI images can be automatically analyzed to recognize visual patterns associated with brain tumors. This technology enables faster, more consistent segmentation, and supports physicians in decision-making.

2.2 Problem Statement

Although MRI is the gold standard for diagnosing brain tumors, tumor identification and segmentation are still performed manually by medical personnel. This process is time-consuming, subject to a high degree of subjectivity, and can potentially lead to inconsistencies between examiners, especially in cases of tumors with complex shapes and boundaries.

Furthermore, the high number of brain tumor cases is not matched by the availability of specialists, risking delays in diagnosis and treatment. Furthermore, despite the rapid advancements in computer vision and artificial intelligence (AI) technologies, their application to the automated segmentation of brain tumors from MRI images still faces challenges in terms of accuracy, generalizability across tumor types, and reliability of the segmentation results.

Therefore, the main problem in this project is how to design and implement an AI model that is capable of segmenting brain tumors automatically, accurately, and consistently from MRI images, so that it can be used as an effective diagnostic tool for medical personnel.

2.3 Objectives

The objectives of this project are as follows:
1. Developing an Artificial Intelligence (AI) based model capable of automatically segmenting brain tumors in MRI images.
2. Produces specific areas on an MRI image that accurately show the location and shape of a brain tumor.
3. Provides a segmentation mask overlay to aid tumor visualization and facilitate interpretation by medical personnel.
4. Improves the efficiency and consistency of the tumor identification process compared to manual analysis methods.
5. Supporting medical personnel as a diagnostic tool, especially in the process of early detection and planning of brain tumor treatment.

2.4 Significance

This project holds significant significance in the fields of health and technology, particularly in the use of Artificial Intelligence (AI) for medical image analysis. By

developing an AI-based brain tumor segmentation model, this project is expected to significantly contribute to improving the efficiency and accuracy of MRI-based diagnostic processes.

From a medical perspective, the resulting segmentation results can help medical personnel identify tumor location and boundaries more clearly and consistently, thus supporting clinical decision-making, therapy planning, and early detection of brain tumors. This model serves as a diagnostic aid, not a substitute for medical personnel, thus improving the overall quality of healthcare services.

From a technical and academic perspective, this project contributes to the application of computer vision and machine learning to real-world healthcare problems. It can also serve as a reference or foundation for further research on medical image segmentation, particularly on various types of brain tumors with varying characteristics.

Furthermore, the implementation of an automated segmentation system has the potential to reduce the workload of medical personnel, speed up MRI analysis, and improve the consistency of interpretation results. Therefore, this project is expected to provide long-term benefits for both the medical world and the development of AI technology in the healthcare sector.

CHAPTER 2
RELATED WORKS

This chapter discusses previous research related to brain tumor segmentation in MRI images using computer vision and machine learning approaches. This review covers various methods and strategies used for automated segmentation, as well as the challenges faced in the medical image analysis process. This review aims to identify research gaps that will form the basis for the development of the model proposed in this project.

Brain tumor segmentation from MRI images has been extensively studied due to its importance in clinical diagnosis. Havaei et al. proposed an automatic segmentation approach using deep learning techniques to address the challenges of complex tumor shapes and heterogeneous appearances in MRI data [3]. The method was evaluated on the BRATS 2013 dataset and assessed using the Dice Similarity Coefficient, demonstrating competitive segmentation performance across different tumor regions. The proposed approach achieved second place in the BRATS 2013 challenge while maintaining relatively efficient processing time. This work demonstrates the effectiveness of learning-based methods for brain tumor segmentation and serves as a foundational reference for subsequent research in medical image analysis.

Ranjbarzadeh et al. provide a comprehensive review of artificial intelligence techniques applied to brain tumor segmentation in MRI images, covering more than 100 research studies published between 2015 and 2022 [4]. The review categorizes methods into supervised, unsupervised, and deep learning approaches, and discusses how these techniques have been used to automate tumor detection and segmentation, addressing challenges such as variability in tumor appearance and limited clinical resources. This work also analyzes the performance of contemporary models,

highlights common datasets and evaluation metrics, and identifies open questions and future research directions in the field of AI-based MRI tumor segmentation.

Ahmed M. Gab Allah et al. proposed the Edge U-Net model for brain tumor segmentation, which integrates boundary information explicitly into the segmentation process to improve localization accuracy [5]. The model was evaluated on a public MRI dataset containing multiple tumor types (glioma, meningioma, pituitary), and achieved Dice similarity scores of approximately 88.8% for meningioma, 91.76% for glioma, and 87.28% for pituitary tumors, demonstrating competitive performance compared to existing segmentation frameworks. This work highlights how incorporating boundary-aware features and a tailored loss function can enhance segmentation precision in automated MRI analysis.

A different paper presents a comprehensive overview of existing MRI-based brain tumor segmentation techniques, from classical image processing methods to more recent deep learning approaches [6]. It examines a range of segmentation strategies, categorizes them (e.g., threshold-based, region-based, model-based), and discusses their relative strengths and limitations in accurately identifying tumor tissues from healthy brain regions. The paper emphasizes that fully automatic methods, including deep learning-enhanced frameworks, provide more reproducible and efficient segmentation compared to manual or semi-automated processes, and notes ongoing challenges such as boundary ambiguity and variability in tumor appearance across patients.

CHAPTER 3
METHODOLOGY

4.1 Dataset

For this project, a dataset of brain tumor patients MRI images and each of its corresponding masks was used to later train the model. The dataset contained various types of brain tumor (glioma, meningioma, and pituitary) from a total of 233 patients [2], [7]. MRI Images from this dataset have varying angles that it was taken from, allowing identification of the tumor from different MRI image angles. The dataset is a public dataset taken from Science Data Bank, comprising of images acquired at Nanfang Hospital, Guangzhou, China, and General Hospital, Tianjin Medical University, China, from 2005 to 2010 using spin-echo-weighted images with a 512×512 matrix [7].

TABLE 1.   Brain Tumor Image Classification

No.	Tumor Type	Number of Image-Mask Pairs
1	Glioma	1,426
2	Meningioma	708
3	Pituitary	930

However, trimming was done before preprocessing as a way to simplify the training for the model. During previous attempts of training the model without dataset trimming it reached large model sizes and lengthy training time. Trimming was done to also ensure a balanced representation of each tumor type class. After trimming of the dataset, each tumor type has 200 image-mask pairs.

4.2　Pre-processing

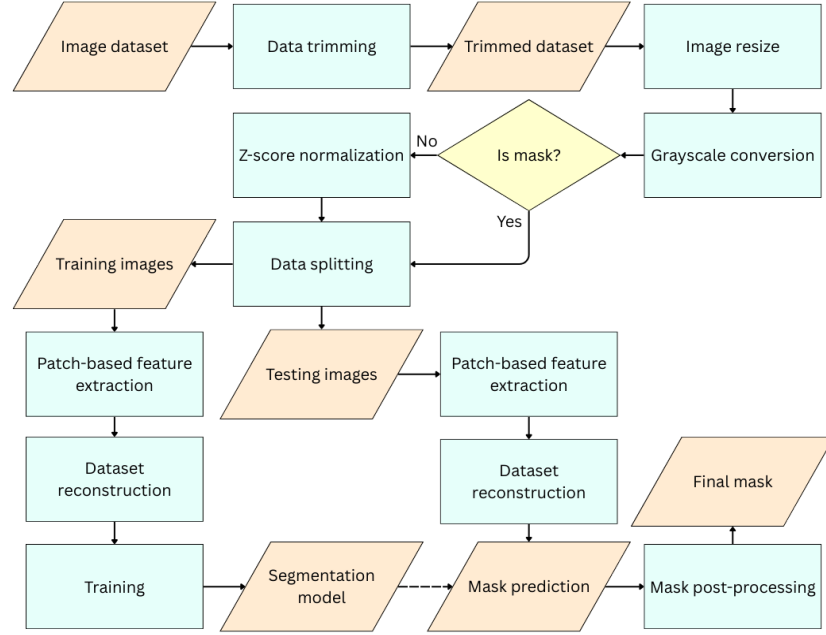The fundamental pre-processing steps are outlined in Fig. 1.

Fig. 1.　Flowchart for the basic training and testing steps

1. **Image processing**

   Image processing begins the pre-processing steps by resizing the image to a fixed size of 256 × 256px to achieve a uniform input. Afterwards, the image gets converted to grayscale to emphasize the features of each image's intensity, especially as MRI images are mostly grayscale. During this process as well, z-score normalization is done to images except for its mask pair, keeping the mean on 0 and standard deviation on 1. Z-score normalization is done to ensure that input with varying brightness settings from different MRI images will not influence the model's predictions [8]. Finally, one-hot encoding is also done to convert the tumor type labels into an equally representative array representation.
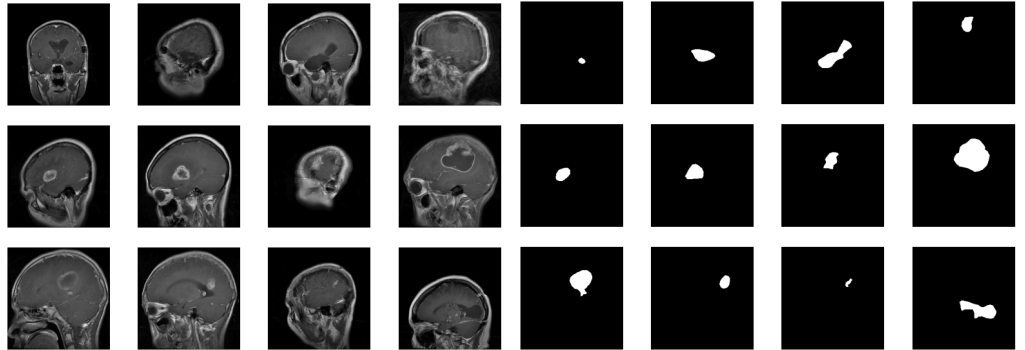

Fig. 2.　Sample of image processing results, MRI images (left) and its mask (right)

2. **Data splitting**

   The dataset is then split into training, validation, and testing sets from the initial trimmed set. The split was done with a ratio of 70:15:15. The split is also done

4

with stratified sampling based on the class of the tumor to maintain the distribution of the classes in each subset. The models in this dataset were trained using the training dataset, the validation dataset is then used to evaluate on unseen data. The testing set was available to use but served the same purpose as the validation set in this case, as both are unseen and are not used to tune the model.

3. **Patch-based feature extraction**
   Since this project uses classical machine learning (ML) to tackle computer vision tasks, manual feature extraction is needed to suffice the model. For image segmentation tasks, patch-based feature extraction is necessary as the output is a mask which is essentially a prediction of each pixel of the image. Images are first broken down into pixels, totaling into 65,536 pixels per image. A padding is then added to the outer part of the image with a size of 3 pixels, each pixel is then compared to its surrounding neighbours in a 7x7 area. The features which are computed in each pixel are: mean of intensity, variance of intensity, and GLCM (contrast, correlation, energy, homogeneity, and entropy). These features are compiled into a feature vector per pixel, which then later gets fed to the model with the label of the mask (0 or 1).

4. **Dataset reconstruction**
   Patch-based feature extraction from the previous step is done on every set (training, validation, and testing). A new dataset is then later formed for each set, containing each pixel's features of the images that originally were in the set. The mask ground truth was also made as labels for the dataset.

4.3   Model Architecture

The model used to tackle this problem is random forest, an ensemble learning model utilizing decision tree classifiers and averaging [9]. The random forest used in this project utilizes 125 trees with the maximum depth of 40. A max_samples value of 0.5 is used and min_samples_leaf of 2 is set as well. A balanced class weight is also set in the parameters to ensure that tumor pixels are not underrepresented.

4.4   Training Setup

In training the random forest model, a multiprocessing system was used on a local machine. The training used 16 concurrent workers, taking a total of 31.6 minutes to complete.

4.5   Evaluation Metrics

To evaluate the model's performance, the model was evaluated with the validation/test dataset. Metrics that were used in evaluating the model are pixel accuracy and F1-score/dice coefficient. In this case as the mask is binary, the F1-score and dice coefficient are equivalent. These metrics are used to also help in tuning for the model's performance.

4.6   Post-processing

After the model's hyperparameters have been tuned, the generated mask images are processed further to get a more interpretable result. Post-processing techniques used in this project include calculating the mask in probability form, smoothing, converting back to binary, and selecting the largest region. Smoothing is done using a gaussian filter kernel size of 5 and a sigma of 2. The smoothened probability map is then converted back to binary form for display. Finally, the largest predicted tumor region is extracted using

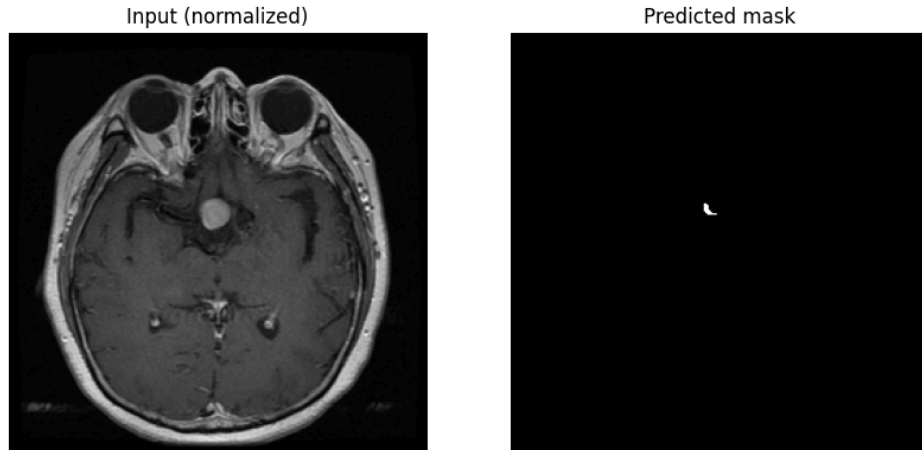OpenCV's connected components to detect and select the largest shape.

Fig. 3. Post processing prediction result of a tumor mask

CHAPTER 4
IMPLEMENTATION & RESULTS

5.1 System Details

The brain tumor segmentation system is implemented as a web application. It utilizes a machine learning model that is integrated into the web interface. The machine learning system consists of MRI image pre-processing, patch-based feature extraction, Random Forest classification, and post-processing to generate the final segmentation mask. Python was used as the primary programming language, with libraries such as OpenCV, NumPy, scikit-image, and Scikit-learn supporting image processing, feature computation, and model training. The system accepts MRI images as input and outputs binary segmentation masks.

The frontend is built using the React framework, utilizing HTML, CSS, and JavaScript to provide a responsive and intuitive user interface for medical professionals. This frontend communicates with a backend powered by a Python Flask API, which serves as the bridge between the user interface and the model. When an MRI image is uploaded, it is transmitted via a RESTful API request to the Flask server, where the core processing occurs. Once complete, the results are sent back to the front end as a result. The results are binary segmentation masks that can be overlaid on the original images to assist interpretation by medical personnel.

5.2 Experiments

Experiments were conducted using the trimmed dataset containing 600 MRI image–mask pairs, evenly distributed across glioma, meningioma, and pituitary tumor classes. The dataset was split into training, validation, and testing sets using a stratified ratio of 70:15:15 to preserve class balance. Pixel-level features extracted through the patch-based approach were used to train the Random Forest model, while hyperparameter tuning was performed based on validation performance. Evaluation was carried out on unseen data to assess the model's generalization capability and segmentation reliability.

5.3 Visualizations

Visualization was used to qualitatively evaluate the segmentation performance of the proposed system by comparing original MRI images, ground truth masks, predicted masks, and overlay results. Post-processing techniques such as Gaussian smoothing and

connected component analysis were applied to reduce noise and retain the most relevant tumor region. These visual results demonstrate the model's ability to localize tumor areas and produce clearer, more interpretable segmentation outputs.
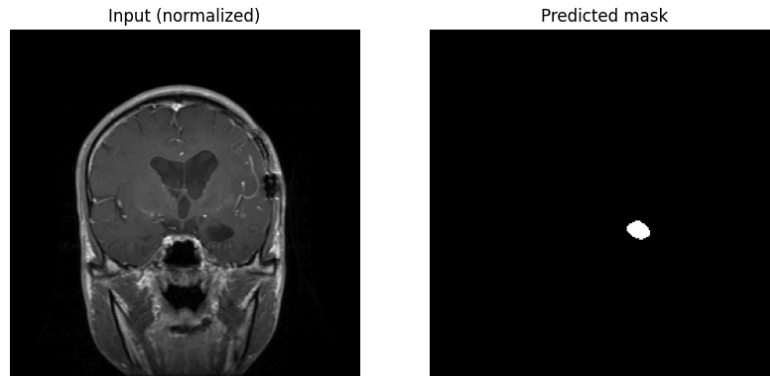

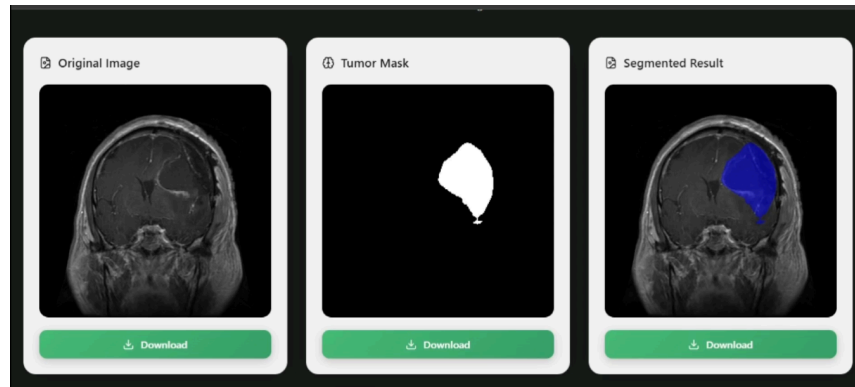Fig. 4. Prediction result visualization of a tumor mask

Fig. 5. Visualization prediction result of a tumor mask on web app

5.4 Results Obtained

The segmentation performance was evaluated using pixel accuracy and F1-score (Dice coefficient), which are suitable metrics for binary segmentation tasks. The model performance is 0.951 for pixel accuracy and 0.952 for Dice/ F1-score. The results indicate that the proposed Random Forest-based approach achieves accurate and consistent tumor segmentation across different MRI images and tumor types. High overlap between predicted masks and ground truth confirms the effectiveness of the feature extraction and post-processing pipeline. Overall, the system shows strong potential as a supportive diagnostic tool that improves efficiency and consistency compared to manual segmentation.

CHAPTER 5
DISCUSSION & LIMITATIONS

6.1 Analysis of Performance

When tested using the validation/test sets, the model demonstrated reasonable performance with 95.14% accuracy and a F1-score or dice coefficient of 95.27%. The model actually slightly has a higher dice coefficient as compared to accuracy. This suggests that the model is actually capturing the tumor regions well, and are not just correctly predicting background. However it is to note that these numbers are calculated before post-processing is done, so it may not be representative of prediction results in the

implementation.

6.2 Challenges

Some challenges were encountered during the making of this project. The main challenge encountered for this project is the complexity of the task for an ML model. Even though to tackle the simpler nature of ML model, an ensemble learning model is used, the task requires extensive dataset preparation including patch-based feature extraction. Patch-based feature extraction itself caused a few troubles through the computational resources it requires. Initially LBP, gabor, and spatial features were intended to be implemented as methods for feature extraction, yet these features combined resulted in really long processing times even with multiprocessing and consumed a very large amount of memory when fed to the model, causing memory failures in the system it is trained on. As a result of patch-based extraction with the abundance of features taken as input for the model, it also generated another challenge when model training was finished. The model is quite heavy in file size when saved, creating file size limit problems with version control in remote repository GitHub.

6.3 Trade-offs

While the model demonstrated efficiency in processing time, the model has trade-offs in some capabilities of predicting the whole tumor region and hidden tumor regions. The model is also rather large in file size, taking up 2,06 GB of space, with the current parameters setup which allow for reasonable accuracy.

CHAPTER 6
CONCLUSION & FUTURE WORKS

This project successfully developed an automated brain tumor segmentation system using a classical machine learning approach based on Random Forest classification and patch-based feature extraction. The proposed system demonstrated strong performance, achieving a pixel accuracy of 95.14% and a Dice coefficient of 95.27% on unseen MRI data, indicating accurate and consistent tumor region segmentation across different tumor types. Through a combination of pre-processing, manually crafted texture features, and post-processing techniques, the system was able to generate interpretable segmentation masks that are able to assist medical personnel in tumor localization and visualization. While the approach suggests effectiveness, it also introduced challenges related to computational cost, memory usage, and large model size due to pixel-level feature extraction. Despite these limitations, the results show that classical machine learning methods remain a viable and reliable alternative for medical image segmentation tasks, particularly as supportive diagnostic tools, and provide a strong foundation for future improvements in efficiency, scalability, and clinical applicability.

Future works may experience with different types of ensemble models, also expand on the usage of the dataset with more computational resources. Other applications which are possible include classifying the tumor type as a task after the tumor has been segmented, to further boost focus on the tumor areas when classifying, although assuming segmentation is reliably covering the tumor regions, which in itself is currently still able to be improved. Improved segmentation performance would also enable the integration of more advanced analysis stages, such as tumor grading, volume estimation, and longitudinal monitoring for treatment evaluation. In addition, future studies may compare classical machine learning approaches with deep learning-based segmentation methods to analyze trade-offs in accuracy, computational cost, and interpretability. By addressing these aspects, the proposed system can be further refined into a more comprehensive and reliable computer-aided diagnosis framework for clinical use.

REFERENCES

[1] D. Putri, Y. Bisri, M. Rasman, and S. Saleh, 'Kemoterapi pada Pasien Operasi Tumor Otak Metastasis: Apa Implikasi Anestesinya?', *Jurnal Neuroanestesi Indonesia*, vol. 8, pp. 50–59, 01 2019.

[2] M. Pichaivel, G. Anbumani, P. Theivendren, and M. Gopal, "An Overview of Brain Tumor," *www.intechopen.com*, Jan. 09, 2022. https://www.intechopen.com/chapters/79979

[3] M. Havaei, A. Davy, D. Warde-Farley, A. Biard, A. Courville, Y. Bengio, C. J. Pal, P.-M. Jodoin, and H. Larochelle, "Brain tumor segmentation with deep neural networks," Medical Image Analysis, vol. 35, pp. 18–31, Jan. 2017, doi: 10.1016/j.media.2016.05.004.

[4] R. Ranjbarzadeh, A. Caputo, E. B. Tirkolaee, S. Jafarzadeh Ghoushchi, and M. Bendechache, "Brain tumor segmentation of MRI images: A comprehensive review on the application of artificial intelligence tools," Computers in Biology and Medicine, vol. 152, Jan. 2023, Art. no. 106405, doi: 10.1016/j.compbiomed.2022.106405.

[5] A. M. Gab Allah, A. M. Sarhan, and N. M. Elshennawy, "Edge U-Net: Brain tumor segmentation using MRI based on deep U-Net model with boundary information," Expert Systems with Applications, vol. 213, Part A, p. 118833, Mar. 2023, doi: 10.1016/j.eswa.2022.118833.

[6] A. Sharma, S. Aggarwal, and H. Kaur, "A review on brain tumor segmentation of MRI images," Magnetic Resonance Imaging, vol. 61, pp. 247–259, Sep. 2019, doi: 10.1016/j.mri.2019.05.043.

[7] J. Cheng, "Brain Tumor Dataset," Scidb.cn, Nov. 2022, doi: 10.57760/sciencedb.06290.

[8] GeeksforGeeks, "ZScore Normalization: Definition and Examples," GeeksforGeeks, Jul. 26, 2024. https://www.geeksforgeeks.org/data-analysis/z-score-normalization-definition-and-examples/

[9] Scikit-Learn, "sklearn.ensemble.RandomForestClassifier — scikit-learn 0.20.3 Documentation," Scikit-learn.org, 2025. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

- **Generative AI usage statement**
  Generative AI was utilized in this project to assist on the ideation of the project, to create an outline of the report, and as audio in the demo video.

- **Team contribution statement**
  Metode kolaborasi kontribusi yang digunakan: Git dan GitHub
  (https://github.com/Garchompz/cv-brain-tumor-classification-segmentation)

Anggota Kelompok	Kontribusi	Detail
Albertus Edbert Chandrajaya	Dataset selection	● Help find valid datasets for our specified theme
	Final Report	● Contribute in creating and completing the final report (abstract, introduction, related works, conclusion)
	Presentation slides creation	● Contributed in making slides for the final presentation
	Final video making	● Participate in presenting material ● Recording the video on Zoom
Kent Amadeo Timotheus	Project theme ideation	● Lead the formulation of the project theme to be taken (model, task)
	Dataset selection	● Help find valid datasets for our specified theme
	Model building	● Researched models suitable for the problem ● Create the dataset pre-processing section ● Perform manual feature extraction using GLCM and intensity ● Reconstruct the dataset containing the extracted features ● Create the modeling section using Random Forest ● Train the model ● Evaluate the model ● Perform hyperparameter tuning to obtain accurate results ● Add post-processing to the model's prediction mask for easier interpretation
	Final video making	● Participate in presenting material ● Recording the video on Zoom
	Report creation	● Create methodology and discussion

		sections of the report ● Help organize the final report
	Presentation slides creation	● Contributed in making slides for the final presentation
	Git contribution	● Perform push and merge during pre-processing and model building
	Team management	● Helps divide tasks and plan project timelines
Theodore Zachary	Project Ideas	● Contributed in helps find project idea
	App Development	● Develop front end using html css and js ● Develop back end using python Flask api and run the AI
	Demo video	● Create a demo video from web app
	Presentation slides creation	● Contributed in making slides for the final presentation
	Final video making	● Participate in presenting material ● Recording the video on Zoom

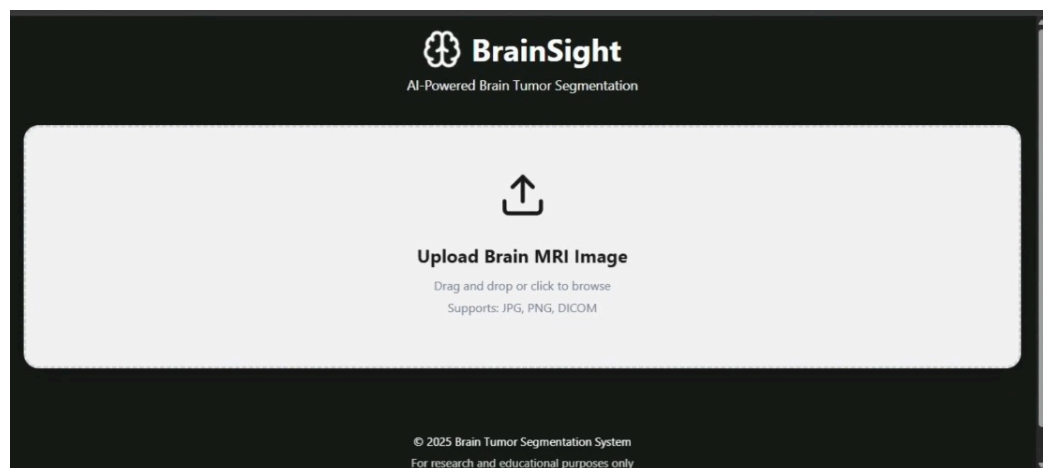● **Screenshots**

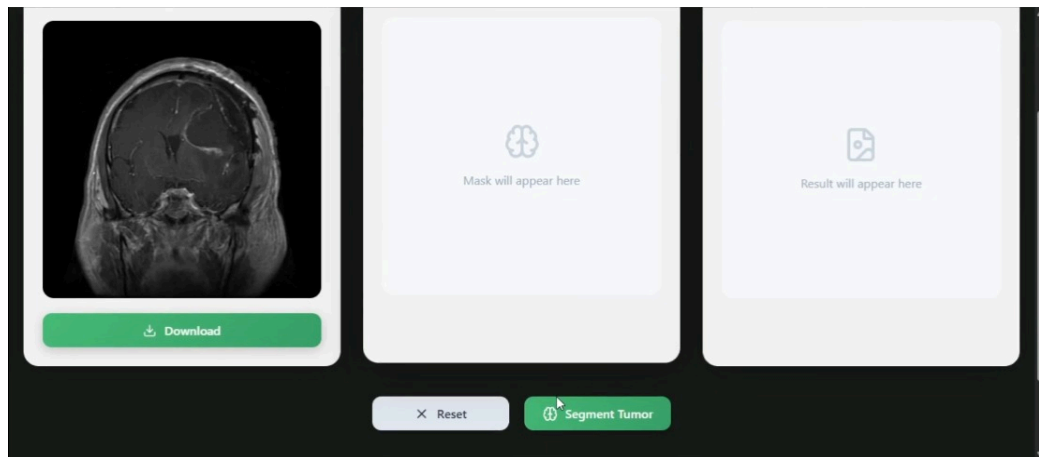Fig. 6.   Implementation web app initial preview

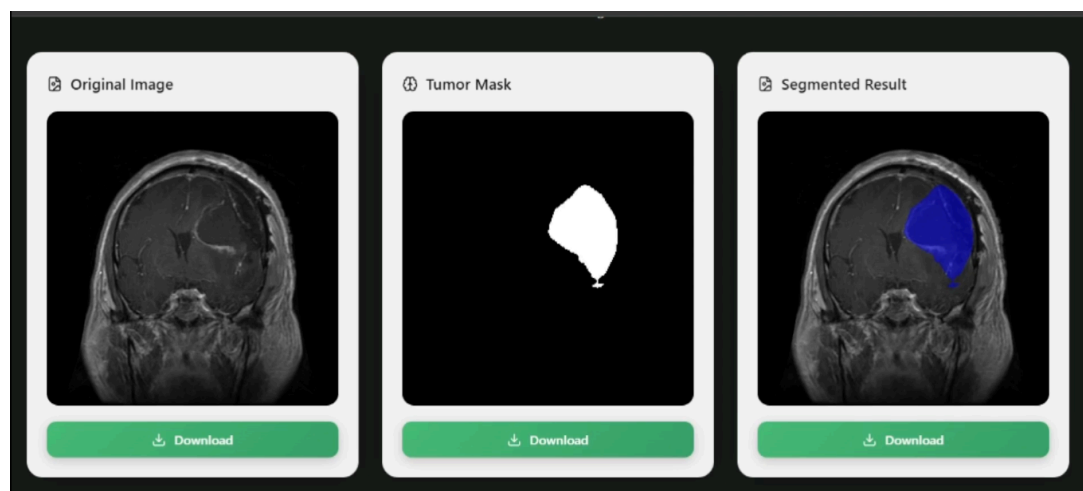Fig. 7.   Implementation web app uploaded image preview

Fig. 8.   Implementation web app segmented image preview

● **Code snippet**

```
rf_seg = RandomForestClassifier(
    n_estimators=125,
    max_depth=40,
    class_weight='balanced',
    min_samples_leaf=2,
    max_samples=0.5,
    random_state=42,
    verbose=1,
    n_jobs=-1
)
```

Fig. 9.   Code snippet for model building