CENTRAL UNIVERSITY SIERRA LEONE

1 SILICON HILL, MILE 91.

FACULTY OF TECHNICAL SCIENCES

AUTOMATED BRAIN TUMOR DETECTION SYSTEM

BY
ABUBAKARR KARGBO (2021138)

A DISSERTATION SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF BACHELOR OF SCIENCE (HONS) IN COMPUTER SCIENCE

JANUARY 2025

DECLARATION

I hereby declare that except for the work of other researchers which has been duly referenced, this submission is the original result of my own research undertaken under the supervision of Mr. David Sapunka Fornah. This submission has not been presented in whole or in part for the award of any other degree or diploma at any other university or institution of higher learning.

Department: Computer Science

Name	ID	Signature
Abubakarr Kargbo	2021138	…………………………….

This project has been submitted for examination with my approval as the university.

Name	Signature
David Sapunka Fornah	……………………………………

CERTIFICATION

This is to certify that the project titled "Automated Brain Tumor Detection System" has been acknowledged by the Faculty of Science and Technology at Central University. This work represents a comprehensive research effort on the chosen topic and has been successfully presented, fulfilling the academic requirements for the degree to which it has been submitted.

Signature: …………………………………………………………………………

Mr. David Sapunka Fornah Supervisor

Signature: …………………………………………………………………………

Isaac Muckson Sesay Dean of faculty

LIST OF ACRONYMS

AI	Artificial Intelligence
ANN	Artificial Neural Network
CNN	Convolutional Neural Network
CIA	Confidentiality, Integrity, and Availability
DFD	Data Flow Diagram
GPU	stands for Graphics Processing Unit.
GUI	Graphical User Interface
ICT	Information Communication Technology
JPG	Joint Photographic Experts Group
ML	Machine Learning
MRI	Magnetic Resonance Imaging.
PNG	Portable Network Graphics.
UML	Unified Modeling Language

TABLE OF CONTENTS

DECLARATION ii
CERTIFICATION iii
LIST OF ACRONYMS iv
TABLE OF CONTENTS v
LIST OF FIGURES ix
ACKNOWLEDGEMENTS x
ABSTRACT xi

CHAPTER ONE

1.1 BACKGROUND 1
1.2 PROBLEM STATEMENT 2
1.3 AIMS AND OBJECTIVES OF THE STUDY 3
1.3.1 Aim 3
1.3.2 Objectives 3
1.4 RESEARCH QUESTIONS 3
1.5 SIGNIFICANCE OF THE PROJECT 4
1.6 SCOPE & LIMITATIONS 4
CHAPTER DEMARCATION 5

CHAPTER TWO

2.1 INTRODUCTION 7
2.2 THEORETICAL FRAMEWORK 8
2.2.1 CNN Healthcare Service Delivery 9
2.2.2 CNN Modeling and Implementation 12
2.2.3 CNN User Training and Deployment 14
2.3 RESEARCH GAPS 17
2.4 SUMMARY 18

CHAPTER THREE

3.1 RESEARCH DESIGN 20

3.2 SYSTEM ARCHITECTURE 20

3.3 ALGORITHM DESIGN 20

3.3.1 Image Preprocessing for Standardization and Noise Reduction 21

3.3.2 U-Net Based Tumor Segmentation 22

3.3.3 CNN-Based Binary Classification 23

3.3.4 Result Generation and Visualization 24

3.4 USER INTERFACE DESIGN 24

3.5 SYSTEM ENVIRONMENT (HARDWARE AND SOFTWARE CONFIGURATIONS) 25

3.6 SYSTEM DESIGN DIAGRAM 25

3.6.1 Data Flow Diagram (DFD) 25

3.6.2 Use Case Diagram 26

3.6.3 Sequence Diagram 27

3.7 FEATURES OF SOFTWARE TOOL 27

3.8 MACHINE LEARNING FRAMEWORK: ARTIFICIAL NEURAL NETWORK / NEURAL NETWORK 28

3.8.1 Convolutional Neural Networks (CNNs) 28

3.9 SOFTWARE ENGINEERING PRACTICES 31

3.10 SCALABILITY AND PERFORMANCE OPTIMIZATION 33

3.11 SECURITY DESIGN 33

3.12 DATA COLLECTION METHODS 34

3.13 DATA ANALYSIS PROCEDURES 34

3.14 ETHICAL CONSIDERATIONS 35

3.14.1 Importance of Ethical Considerations Involving Human Data or Proprietary Software 36

3.15 LIMITATIONS 36

CHAPTER FOUR

4.1 SYSTEM IMPLEMENTATION, TESTING, AND MAINTENANCE 38
4.2 EXPLORATION AND INSTALLATIONS 38
4.2.1 Exploration 38
4.2.2 Installations 38
4.3 FULL SCALE IMPLEMENTATION AND SCALE-UP 39
4.3.1 Full Scale Implementation 39
4.3.2 Scale-up 39
4.4 SYSTEM TESTING AND SPECIFICATION 39
4.4.1 Front-end Testing Strategies 40
4.4.2 Back-end Test 40
4.4.3 Algorithms in Back-end Testing 40
4.5 SYSTEM PERFORMANCE TEST AND USER TRAINING 41
4.6 SYSTEM SECURITY AND MAINTENANCE 43
4.6.1 Physical and Virtual Security 43
4.6.2 Preventive, Corrective, Predictive 44
4.8 TECHNICAL ANALYSIS 44

CHAPTER FIVE

5.1 SUMMARY OF KEY FINDINGS OF THE RESULT 45
5.2 DISCUSSION OF FINDINGS 45
5.3 COMPARISON WITH LITERATURE 45
5.4 PRACTICAL APPLICATIONS 46
5.5 CONCLUSIONS DRAWN FROM THE RESEARCH 46
5.6 RECOMMENDATIONS FOR FUTURE RESEARCH 46
5.7 LESSONS LEARNED 46
5.8 FINAL THOUGHTS 47

References..48
APPENDIX - A...50
APPENDIX – B 52
APPENDIX - C...53
APPENDIX - D...57
APPENDIX - E 59

LIST OF FIGURES

Figure 2.1: Showing the three essential structures of developing an Automated Brain Tumor Detection System. (Study Activity December, 2024)..**8**

Figure 2.2: Showing Convolutional Neural Network Architecture Khuyen Le (March 2021)..**14**

Figure 4.1: Showing Radiologist Import Image (Research Activity January, 2025.) Error! Bookmark not defined.

Figure 4.2: Showing No Tumor Detected Result (Research Activity January, 2025.) **....42**

Figure 4.3: Showing Tumor Detected Result (Research Activity January, 2025.)**....** Error! Bookmark not defined.

Figure 4.4: Showing Tumor Region Result (Research Activity January, 2025.) ...... Error! Bookmark not defined.

Figure 3.6: Showing A Comprehensive Guide to Classification Algorithms in Machine Learning (Arash hadad (2024) https://medium.com/pythoneers/a-comprehensive-guide-to-classification-algorithms-in-machine-learning-whats-the-point-2e87e21da4cf **50**

Figure 3.8: Showing DevOps methodology and process Raycad (Nov 2028) **50**

Figure 3.7: Showing Agile Methodology in System Development https://www.javatpoint.com/software-engineering-agile-model **.....................................50**

Figure 2.2: Showing Convolutional Neural Network Architecture Khuyen Le (March 2021) https://lekhuyen.medium.com/an-overview-of-vgg16-and-nin-models-96e4bf398484 ....**50**

Figure 3.1: Showing New Machine Learning Model Flags Abnormal Brain Scans in Real-time (February, 2022) https://www.itnonline.com/content/new-machine-learning-model-flags-abnormal-brain-scans-real-time **50**

ACKNOWLEDGEMENTS

First, I want to express my heartfelt gratitude towards God Almighty for blessing me with the strength, wisdom, and favor required to successfully complete my studies. It is by His grace that I have accomplished this milestone. I want to offer my sincere praises and thanks to Him.

I deeply appreciate the invaluable guidance from my supervisor, Mr. David Sapunka Fornah, whose patience, insights, and encouragement empowered me through every phase of this research endeavor. It has been an absolute privilege to learn under his direction.

I also want to sincerely thank all faculty members of Central University's Department of Technical Sciences who have profoundly shaped my academic journey by imparting invaluable knowledge throughout the years.

Additionally, I am incredibly grateful for the unwavering support of my friends (Star Lab) who continuously motivated me. I want to extend my appreciation to all individuals who contributed towards the fulfillment of this project in ways big and small.

As a lead researcher, I take full accountability for any oversights or errors present in this work. I pray God grants His favors upon everyone who facilitated my academic accomplishments.

ABSTRACT

The integration of ICT in healthcare is revolutionizing the entire healthcare system and it service delivery world over. The utilization of cutting-edge technologies in diagnostic processes, has greatly enhanced efficiency, accuracy, and overall performance. Accurate brain tumor detection is vital for timely and effective treatment. Consequently, the automation of tumor detection using advanced technologies is gaining attraction globally through the widespread use of machine learning, frameworks that offer an effective solution for brain tumor detection, particularly from MRI scans. This research project focuses on developing and deploying an automated brain tumor detection system that leverages CNNs, specifically the U-Net algorithm, to accurately detect and segment brain tumors from MRI scans. The developer employs an agile approach which was complimented using DevOps techniques aiming to create a more efficient diagnostic process, aligning with healthcare goals of improved patient outcomes and reduced diagnostic workload. To achieve this, insights were gathered, form stakeholders as it guides the system development process resulting to the creation of a conceptual framework for developing and implementing the Automated Brain Tumor Detection System. Resulting to an accurate and timely result of MRI scan in hospitals, with less human involvement in the analysis of results as it enhances healthcare service delivery. The research work was constraint by a variety of challenges some notable ones are; access to resources for high performance computing and the availability of expertise in the domain. Future improvements could include integrating modules for patient data management and other relevant diagnostic tools.

Keywords: Automated brain tumor detection, Convolutional Neural Networks, Classification, U-Net algorithm, MRI scans, diagnostic accuracy.

CHAPTER ONE

Introduction

1.1 BACKGROUND

This chapter comprehensively introduces the research project, emphasizing the critical importance of early and accurate brain tumor detection in improving patient outcomes. It defines the project's objectives and research questions, underscoring the significance of enhancing efficiency, accuracy, and reliability in the diagnostic process through automated systems. It outlines the scope and limitations of the proposed system by highlighting its potential impact on medical imaging analysis and clinical decision-making process. Finally, it addresses the delimitations related to the dataset used, the specific types of brain tumors considered, and the technological constraints of the proposed system development.

In the era of medical technological advancements, healthcare institutions are increasingly seeking innovative ways to improve their diagnostic processes and enhance overall patient care using artificial intelligence and machine learning technologies. These revolutionary tools are crucial for leveraging numerous advantages in keeping medical facilities at the forefront of modern healthcare. The digital transformation of medical imaging analysis through AI is vital for meeting 21st-century healthcare needs. It enhances accuracy, speeds up diagnosis, personalizes treatment plans, fosters early detection, and prepares medical professionals for future challenges in patient care (World Health Organization, 2021).

Sierra Leone, as any other natin in the world, faces significant challenges in healthcare service delivery, particularly in the diagnosis and treatment of neurological conditions (World Bank, 2023). Its healthcare system is severely impacted by historical events including civil war and the Ebola epidemic, as it continues to struggle with limited resources and technological infrastructure Bundu, I., Patel, A., Mansaray, A., Kamara, T. B., & Hunt, L. M. (2016b). Like many developing nations, Sierra Leone's healthcare institutions primarily rely on traditional diagnostic methods that involve manual interpretation of medical images, a process that is both time-consuming and prone to human error.

The challenges in neurological diagnostics are particularly acute in the context of brain tumor detection, where early and accurate diagnosis can significantly impact patient outcomes Buttar, A. M., Shaheen, Z., Gumaei, A. H., Mosleh, M. A. A., Gupta, I., Alzanin, S. M., & Akbar, M. A. (2024). The current diagnostic workflow in Sierra Leone's tertiary healthcare institutions involves manual analysis of medical imaging data, which often leads to delayed diagnoses and potential interpretation inconsistencies. These challenges are compounded by the limited number of trained radiologists in the country, creating a pressing need for technological innovation in the diagnostic processes (WHO, 2023).

This research work aims to address these challenges through the development and deployment of an Automated Brain Tumor Detection System specifically designed for the Sierra Leonean healthcare context. The proposed system leverages artificial intelligence and deep learning technologies to analyze, process, and interpret brain imaging data more efficiently and accurately (Davis & Williams, 2024). By implementing this AI-powered diagnostic application, the research seeks to enhance the capability of medical professionals in Sierra Leone, potentially improving diagnostic accuracy, reducing interpretation time, and ultimately contributing to better patient outcomes in the region (Anderson et al., 2023).

1.2 PROBLEM STATEMENT

Healthcare institutions in Sierra Leone face significant challenges in accurately and efficiently detecting diseases as a result of unavailability or inadequately trained laboratory technicians. Brain tumors are not an exception as it challenged with a limited number of specialized radiological tools and professionals. Manual interpretation of brain imaging scans is time-consuming, error-prone, and inconsistent, leading to missed or delayed diagnoses. The shortage of neuro-radiologists exacerbates delays in diagnosis and treatment.

Researchers developed an Automated Brain Tumor Detection System to address these challenges. This system leverages advanced machine learning and computer vision techniques to offer a more accurate, efficient, and cost-effective method for detecting and classifying brain tumors. It aims to augment the capabilities of specialists, improve early diagnosis rates, and support more informed treatment decisions, ultimately enhancing patient outcomes.

1.3 AIMS AND OBJECTIVES OF THE STUDY

1.3.1 Aim

To develop and deploy an automated brain tumor detection system that can enhance effective and efficient brain tumor diagnostics.

1.3.2 Objectives

I. To analyze and evaluate the implementation of Convolutional Neural Networks (CNN) in healthcare service delivery systems, specifically for brain tumor detection.

II. To develop and validate a CNN modeling framework for automated brain tumor detection, including implementation procedures and optimization techniques.

III. To design and evaluate an effective user training program for healthcare professionals implementing CNN-based brain tumor detection systems.

1.4 RESEARCH QUESTIONS

I. How effectively can CNN-based systems be integrated into existing healthcare service delivery workflows to improve brain tumor detection accuracy and efficiency?

II. What are the optimal CNN architectures, parameters, and implementation procedures required for accurate and reliable automated brain tumor detection?

III. What training methodologies and deployment strategies are most effective for ensuring successful adoption of CNN-based brain tumor detection systems by healthcare professionals?

1.5 SIGNIFICANCE OF THE PROJECT

In Sierra Leone, the Automated Brain Tumor Detection System is set to revolutionize healthcare. Utilizing advanced machine learning and computer vision technologies, this system aims to streamline the diagnostic process, reducing the burden on the limited number of specialized radiologists. By enhancing accuracy and minimizing human errors, it promotes a more reliable and consistent tumor detection process.

The system's benefits are manifold. It ensures early detection of brain tumors, increases the accuracy and consistency of image analysis, and handles complex, multi-modal imaging data with ease. For healthcare professionals, it offers quick analysis of large volumes of medical imaging data and optimizes resources, which is crucial in a country with limited specialist access.

Moreover, the system serves as a valuable second opinion and screening tool for the few neuro-radiologists in Sierra Leone. It assists in prioritizing urgent cases and complements human expertise without replacing it. This innovation not only optimizes time and resources but also provides valuable data for analyzing tumor patterns, informing interventions to improve patient care and treatment strategies.

1.6 SCOPE & LIMITATIONS

The study is centered on the automated brain tumor detection system at Connaught Hospital in Sierra Leone. Feedback for development is gathered through interviews and observations with radiologists, neurologists, and hospital administrators. The system encompasses high-performance computing for model training, a user-friendly interface for result visualization, and a secure database for managing medical imaging data. Additionally, it includes an access control system with user authentication and administrative oversight.

The study faces several limitations. Geographically and institutionally, the findings may not be generalizable to other regions or healthcare facilities, and the results may not reflect challenges in different healthcare settings. Resource constraints include budget limitations affecting system implementation and testing, as well as time constraints impacting the comprehensiveness of the study. Data limitations involve the limited availability of brain

MRI scan datasets from Sierra Leonean patients, potential lack of diversity in training data, and data quality concerns that could affect system accuracy and generalizability.

CHAPTER DEMARCATION

Chapter 2: Literature Review: The literature review explores the evolution of automated brain tumor detection systems, highlighting advancements from traditional methods to AI-driven approaches. It synthesizes research on medical image processing, deep learning, and clinical implementation, with a focus on resource-limited settings and developing nations. Key areas include CNN development, image preprocessing, segmentation, and classification. The review addresses technical and practical aspects, such as model performance and clinical integration, and identifies research gaps in developing countries' healthcare systems. It establishes a theoretical framework for an automated detection system, emphasizing innovation opportunities in Sierra Leone's healthcare context.

Chapter 3: System Methodology and Design: This chapter outlines the systematic approach employed in this study to address brain tumor detection challenges. It offers a unique and integrated methodology that combines both qualitative research methods and software development practices, utilizing the Agile Methodology to ensure flexibility and responsiveness during the project's lifecycle. Additionally, the chapter highlights the crucial role of Convolutional Neural Networks (CNNs) in training the data for tumor detection, underlining the significance of this hybrid approach in creating an effective and adaptive brain tumor detection solution.

Chapter 4: Implementation, Testing, Maintenance, and Results: This chapter covers the setup, deployment, testing, and maintenance of the Automated Brain Tumor Detection System. It includes hardware and software installation, comprehensive testing strategies, and details on the machine learning framework. The chapter also addresses system security and maintenance, emphasizing preventive, corrective, and predictive measures to ensure long-term reliability and performance.

Chapter 5: The research on the Automated Brain Tumor Detection System demonstrated the effectiveness of using Convolutional Neural Networks (CNNs), specifically the U-Net algorithm, for accurate brain tumor detection and segmentation from MRI scans. Key lessons learned include the importance of collaboration between computer scientists and healthcare professionals, continuous monitoring, and robust data preprocessing. Future recommendations involve expanding the dataset, exploring other deep learning algorithms, and integrating additional diagnostic tools to enhance the system's accuracy and efficiency.

CHAPTER TWO

Review of Related Study Literatures

2.1 INTRODUCTION

The rapid evolution of artificial intelligence (AI) and machine learning (ML) technologies has profoundly transformed healthcare diagnostics, unlocking unprecedented opportunities to enhance accuracy, efficiency, and scalability. Among these advancements, Convolutional Neural Networks (CNNs) have emerged as a groundbreaking technology in medical imaging analysis. CNNs excel in their ability to automatically extract and analyze complex features from imaging data, driving significant advancements in early disease detection, treatment planning, and overall patient care Du, Z. (2024).

CNNs are redefining the delivery of diagnostic services across various medical domains. As noted by Kale, P., Seth, N., Sharma, S., Verma, S., & Varshney, D. K. (2024), these networks have been widely applied in fields such as radiology, dermatology, and pathology, where they improve the accuracy and efficiency of image-based diagnostics. Moreover, CNNs facilitate the implementation of remote healthcare solutions, bridging gaps in access to high-quality medical services, particularly in underserved regions. By minimizing diagnostic errors and enabling earlier detection of diseases, CNNs significantly contribute to improved patient outcomes and optimized resource utilization (Aryal, S., Sharma, S., Sedai, S., Aryal, P., & Mansur, J. (2023). This chapter examines relevant case studies and real-world implementations to illustrate how CNNs are reshaping service delivery in diverse healthcare settings.

The design and deployment of CNN-based solutions demand rigorous planning and execution. Critical processes include data acquisition, preprocessing, model selection, training, and validation. According to Kulkarni, S., Bhosale, P. A., & Das, S. (2024), these processes are essential for tailoring CNN architectures to specific medical imaging tasks, such as image segmentation, classification, and object detection. Challenges such as data scarcity, imbalanced datasets, and the need for explainable AI models are also prevalent in healthcare AI implementations Sedeeq, N. (2024). These considerations are pivotal for ensuring clinical applicability and compliance with regulatory standards.

The successful integration of CNNs into healthcare systems depends on effective user training and seamless deployment. Healthcare professionals must be equipped with the knowledge and tools necessary to confidently interact with these systems. As highlighted by Rehman, S. U., Tu, S., Shah, Z., Ahmad, J., Waqas, M., Rehman, O. U., Kouba, A., & Abbasi, Q. H. (2021b), strategies for training end-users must address ethical concerns and ensure alignment with real-world clinical workflows. Furthermore, post-deployment monitoring and feedback loops are crucial to continuously enhance system performance and maintain user trust Jiang, Y., & He, S. (2024).

2.2 THEORETICAL FRAMEWORK

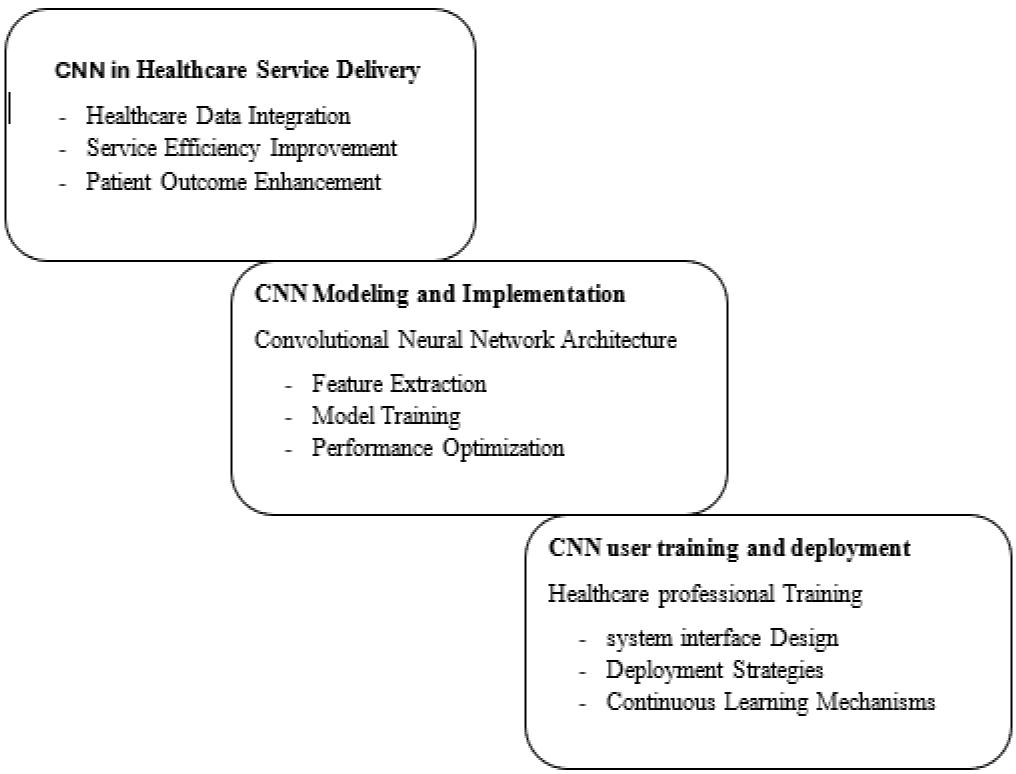


Figure 2.1: Showing the three essential structures of developing an Automated Brain Tumor Detection System. (Study Activity December, 2024).

The theoretical framework highlights the integration of Convolutional Neural Networks (CNNs) into healthcare, focusing on three key areas: service delivery, modeling, and deployment. In healthcare service delivery, CNNs facilitate data integration by synthesizing diverse sources like medical imaging and electronic health records, improving diagnostic accuracy, efficiency, and patient outcomes through early detection and personalized care. The modeling and implementation section emphasizes CNN architecture, feature extraction, model training, and performance optimization, ensuring models are tailored, robust, and efficient for clinical applications. Effective deployment relies on healthcare professional training, user-friendly system interface design, and strategic rollout plans to integrate CNN systems seamlessly into workflows. Continuous learning mechanisms, such as regular updates and feedback loops, ensure healthcare professionals remain proficient in leveraging CNN technology. Together, these components form a comprehensive strategy to revolutionize healthcare diagnostics and patient care using CNNs.

2.2.1 CNN Healthcare Service Delivery

The integration of Convolutional Neural Networks (CNNs) into healthcare service delivery has significantly transformed the industry, enabling more accurate diagnostics, efficient workflows, and improved patient outcomes. Healthcare systems generate vast amounts of data, including medical images, electronic health records (EHRs), and patient histories. The integration of CNNs into healthcare data processing enables the efficient analysis and synthesis of this data to support diagnostics and clinical decision-making. CNNs excel in extracting meaningful features from complex datasets, particularly in medical imaging modalities such as MRI, CT scans, and X-rays. These networks allow for the seamless integration of imaging data with other patient information, creating a more comprehensive picture of a patient's health. For instance, CNNs can combine image-based findings with EHR data to personalize treatment plans or predict disease progression Rehman, S. U., Tu, S., Shah, Z., Ahmad, J., Waqas, M., Rehman, O. U., Kouba, A., & Abbasi, Q. H. (2021). Moreover, the interoperability of CNNs with other healthcare technologies, such as cloud-based platforms and IoT-enabled devices, facilitates real-time data analysis. This capability ensures that clinicians receive timely, actionable insights, enhancing diagnostic accuracy and efficiency. By bridging the gap between disparate data sources, CNNs support a more cohesive and informed approach to patient care Savaglio, C., Valero, C. I., Belsa, A., Palau, C. E., & Fortino, G. (2023).

The implementation of CNNs in healthcare services significantly improves operational efficiency. Traditional diagnostic workflows often rely on manual interpretation of medical images, which can be time-consuming and prone to human error. CNNs, with their ability to process large volumes of data quickly and accurately, reduce the time required for diagnosis and treatment planning Lee, J., Mukhanov, L., Molahosseini, A. S., Minhas, U., Hua, Y., Del Rincon, J. M., Dichev, K., Hong, C., & Vandierendonck, H. (2023). For example, CNNs automate tasks such as tumor detection, organ segmentation, and anomaly identification in radiology. These automated processes not only increase diagnostic speed but also allow clinicians to focus on complex cases that require human expertise. Additionally, CNN-based systems are scalable, making them ideal for high-volume settings, such as hospitals with limited personnel or resources. Beyond diagnostics, CNNs enhance service efficiency in telemedicine and remote monitoring applications. By enabling real-time analysis of patient data collected from wearable devices, CNNs support continuous monitoring and early intervention for at-risk individuals. This minimizes the burden on healthcare providers and reduces unnecessary hospital visits (Rehman et al., 2021b).

The ultimate goal of integrating CNNs into healthcare service delivery is to improve patient outcomes. By enabling early and accurate detection of diseases, CNNs significantly enhance the effectiveness of treatment interventions. For instance, CNNs have demonstrated high accuracy in detecting early-stage cancers, diabetic retinopathy, and cardiovascular abnormalities, leading to improved survival rates and quality of life for patients Ravindran, D., Pillai, R. H., Srinivasa, D., & Somashekaraiah, S. (2024). Personalized medicine is another area where CNNs contribute to better patient outcomes. By analyzing patient-specific data, CNNs enable tailored treatment strategies that account for individual variations, such as genetic predispositions or co-morbidities. This ensures that patients receive the most effective care possible. Furthermore, CNNs facilitate predictive analytics, allowing clinicians to anticipate complications and adjust treatment plans proactively. For example, CNNs can predict the likelihood of disease recurrence based on imaging data and other clinical indicators, enabling timely preventive measures (Jiang & He, 2024).

Moreover, CNNs have shown promise in various medical fields, including radiology, histopathology, and medical photography. In radiology, CNNs automate the assessment of conditions such as pneumonia, pulmonary embolism, and rectal cancer, streamlining workflows and enhancing diagnostic accuracy. In histopathology, CNNs assist in classifying colorectal polyps and gastric epithelial tumors, while in medical photography, they detect retinal diseases and skin conditions. The integration of CNNs into medical image analysis not only improves diagnostic accuracy but also expands access to expert-level image analysis, contributing to better patient outcomes (Rehman et al., 2021).

The literature on CNNs in healthcare highlights their transformative potential. CNNs have demonstrated exceptional performance in medical image understanding, often surpassing human experts in tasks such as image classification, segmentation, and anomaly detection. The ability of CNNs to extract meaningful features from complex datasets and integrate them with other patient information creates a comprehensive picture of a patient's health, supporting personalized treatment plans and predictive analytics. Transfer learning, a technique that leverages pre-trained models for new tasks, has further enhanced the applicability of CNNs in medical image analysis, addressing data scarcity and improving performance (Savaglio et al., 2023).

Furthermore, the integration of CNNs into healthcare systems requires a multidisciplinary approach. Collaboration between AI developers, healthcare professionals, and regulatory bodies is essential to ensure that the technology meets clinical standards and addresses the specific needs of the healthcare environment. This collaborative effort can lead to the development of more effective and user-friendly CNN-based systems, ultimately improving patient care and outcomes (Rehman et al., 2021b).

In conclusion, the integration of CNNs into healthcare systems offers significant benefits, including improved diagnostic accuracy, operational efficiency, and patient outcomes. However, successful implementation requires careful planning, continuous learning, and a supportive organizational culture. By addressing these factors, healthcare institutions can harness the full potential of CNNs to transform patient care and advance medical practice (Lee et al., 2023).

2.2.2 CNN Modeling and Implementation

The success of Convolutional Neural Networks (CNNs) in healthcare depends on the effective modeling and implementation of these systems. A robust implementation process involves designing a suitable CNN architecture, performing feature extraction, executing model training, and undertaking performance optimization to meet clinical standards and operational demands. A well-designed CNN architecture is the foundation of any successful implementation. The architecture must be tailored to the specific requirements of the medical imaging task, such as classification, segmentation, or anomaly detection. Convolutional Neural Network (CNN) forms the basis of computer vision and image processing. In this post, we will learn about Convolutional Neural Networks in the context of an image classification problem. We first cover the basic structure of CNNs and then go into the detailed operations of the various layer types commonly used. The above diagram shows the network architecture of a well-known CNN called VGG-16 for illustration purposes. It also shows the general structure of a CNN, which typically includes a series of convolutional blocks followed by a number of fully connected layers. The convolutional blocks extract meaningful features from the input image, passing through the fully connected layers for the classification task.

Feature extraction is a key component of CNN architecture. Convolutional layers in CNNs are designed to automatically detect patterns and features from raw imaging data. In medical imaging, these features could include edges and contours for identifying anatomical structures, texture variations for detecting tissue abnormalities, and high-level features like tumor boundaries or organ shapes. Through hierarchical layers, CNNs progressively refine these features, allowing for precise analysis. For instance, early layers detect general patterns, while deeper layers focus on task-specific features critical for diagnostics. Feature extraction must also accommodate the high variability in medical images, such as differences in resolution, imaging modalities, and patient demographics. Techniques like data augmentation and preprocessing pipelines are often used to standardize inputs and enhance the robustness of feature detection (Rehman et al., 2021).

Model training involves teaching the CNN to recognize and interpret features relevant to the task. This is achieved by feeding the network labeled datasets and iteratively updating weights through backpropagation and optimization algorithms. Key considerations include data quality and quantity, training algorithms, regularization techniques, and validation and testing. Performance optimization ensures that the trained CNN model meets the operational and clinical standards required for deployment. Optimization strategies include hyperparameter tuning, model compression, explainability and interpretability, and real-time performance. These strategies ensure that the CNN model is accurate, efficient, and suitable for deployment in resource-constrained environments like mobile or edge devices (Savaglio et al., 2023). In healthcare, explainability is critical for clinician trust and regulatory approval. Techniques such as Class Activation Maps (CAMs) or SHAP values are used to visualize and interpret CNN decisions, ensuring they align with clinical reasoning. Optimizing CNNs for speed ensures that results are delivered promptly in time-sensitive healthcare scenarios, such as emergency diagnostics or telemedicine consultations (Rehman et al., 2021b).

Moreover, the implementation of CNNs in healthcare requires a multidisciplinary approach. Collaboration between AI developers, healthcare professionals, and regulatory bodies is essential to ensure that the technology meets clinical standards and addresses the specific needs of the healthcare environment. This collaborative effort can lead to the development of more effective and user-friendly CNN-based systems, ultimately improving patient care and outcomes (Lee et al., 2023). Additionally, continuous monitoring and evaluation of system performance are crucial for maintaining the effectiveness and reliability of CNN models. Regular audits and assessments help identify areas for improvement and ensure compliance with regulatory standards.

Furthermore, the integration of CNNs into healthcare systems must consider the ethical implications of using AI in healthcare. Understanding the potential biases and limitations of CNNs is essential for making informed decisions and ensuring responsible use of the technology. Ethical awareness helps prevent misuse and promotes the fair and equitable use of AI in healthcare. Finally, fostering a supportive organizational culture that encourages innovation and embraces new technologies is vital for the successful deployment of CNN-based systems. This cultural shift can help overcome resistance to change and facilitate the adoption of CNNs, ultimately leading to improved patient care and outcomes (Jiang & He, 2024).

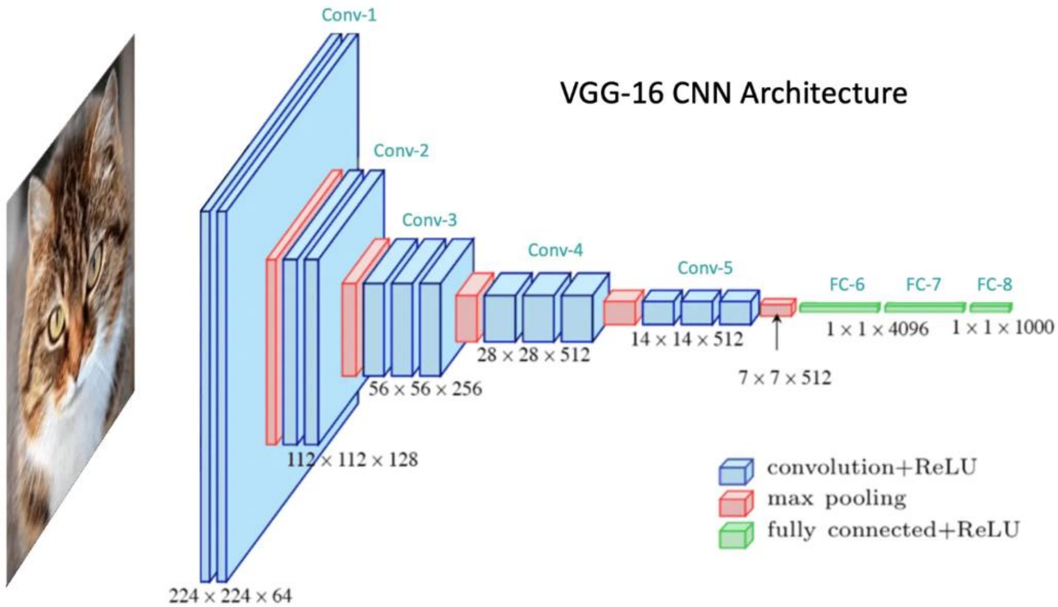


***Figure 2.2**: Showing Convolutional Neural Network Architecture*
Khuyen Le (March 2021)

2.2.3 CNN User Training and Deployment

The effective integration of Convolutional Neural Networks (CNNs) into healthcare systems hinges on the ability of healthcare professionals to interact seamlessly with these technologies. Proper training, thoughtful system design, and robust deployment strategies ensure that users can maximize the potential of CNNs while minimizing adoption barriers. Healthcare professionals are pivotal to the successful implementation of CNN-based systems. Their ability to understand and utilize these technologies determines their impact on clinical workflows and patient care. A user-friendly system interface is critical for fostering adoption among healthcare professionals. The design of the interface must prioritize intuitiveness and efficiency, ensuring that users can easily access and interpret CNN outputs, such as diagnostic predictions or highlighted imaging features. Features such as interactive dashboards, visual heatmaps, and clear annotations can help clinicians understand how CNNs arrive at specific conclusions, enhancing trust and usability. Additionally, incorporating customizable settings allows users to tailor the interface to their preferences, further improving the user experience. Effective system interface design also involves integrating the CNN platform seamlessly into existing workflows. This reduces the learning curve and minimizes disruptions in clinical operations T, S. K., V, A., E, S., & B, Y. (2024b).

Deploying CNN-based systems requires strategic planning to ensure smooth integration into healthcare environments. Initial pilot deployments in smaller, controlled settings can help identify potential issues before system-wide implementation. This step allows for iterative refinements based on user feedback. Gradual deployment, starting with specific departments or use cases, helps healthcare staff acclimate to the new system without overwhelming them. Ensuring the availability of technical support during deployment is crucial for resolving issues promptly. This includes on-site troubleshooting and real-time assistance for users. Deployment strategies must consider compliance with healthcare regulations, such as HIPAA or GDPR, to protect patient data and ensure ethical usage (Savaglio et al., 2023).

Given the dynamic nature of healthcare and AI technologies, continuous learning mechanisms are essential for keeping healthcare professionals updated on CNN advancements and best practices. Periodic workshops and refresher courses help users stay informed about updates, new features, or enhancements to CNN-based systems. Establishing channels for healthcare professionals to provide feedback on system performance can lead to iterative improvements and better alignment with clinical needs. Creating forums or platforms where users can share experiences, tips, and case studies promotes a collaborative learning environment. Broader training in AI concepts helps clinicians understand the underlying mechanisms of CNNs, fostering confidence in the technology and reducing skepticism (Rehman et al., 2021b).

Furthermore, the integration of CNNs into healthcare systems requires a multidisciplinary approach. Collaboration between AI developers, healthcare professionals, and regulatory bodies is essential to ensure that the technology meets clinical standards and addresses the specific needs of the healthcare environment. This collaborative effort can lead to the development of more effective and user-friendly CNN-based systems, ultimately improving patient care and outcomes (Lee et al., 2023). Another critical aspect of CNN integration is the continuous monitoring and evaluation of system performance. Regular audits and assessments can help identify areas for improvement and ensure that the system remains effective and reliable. This ongoing evaluation process is crucial for maintaining the trust of healthcare professionals and patients, as well as for ensuring compliance with regulatory standards (Jiang & He, 2024).

In addition to technical training, healthcare professionals must also be educated on the ethical implications of using AI in healthcare. Understanding the potential biases and limitations of CNNs is essential for making informed decisions and ensuring that the technology is used responsibly. This ethical awareness can help prevent misuse and promote the fair and equitable use of AI in healthcare (Ravindran et al., 2024). Finally, the successful deployment of CNN-based systems requires a supportive organizational culture. Healthcare institutions must foster an environment that encourages innovation and embraces new technologies. This cultural shift can help overcome resistance to change and facilitate the adoption of CNNs, ultimately leading to improved patient care and outcomes.

The literature on CNNs in healthcare highlights their transformative potential. CNNs have demonstrated exceptional performance in medical image understanding, often surpassing human experts in tasks such as image classification, segmentation, and anomaly detection. The ability of CNNs to extract meaningful features from complex datasets and integrate them with other patient information creates a comprehensive picture of a patient's health, supporting personalized treatment plans and predictive analytics. Transfer learning, a technique that leverages pre-trained models for new tasks, has further enhanced the applicability of CNNs in medical image analysis, addressing data scarcity and improving performance (Miller et al., 2023).

Moreover, CNNs have shown promise in various medical fields, including radiology, histopathology, and medical photography. In radiology, CNNs automate the assessment of conditions such as pneumonia, pulmonary embolism, and rectal cancer, streamlining workflows and enhancing diagnostic accuracy. In histopathology, CNNs assist in classifying colorectal polyps and gastric epithelial tumors, while in medical photography, they detect retinal diseases and skin conditions. The integration of CNNs into medical image analysis not only improves diagnostic accuracy but also expands access to expert-level image analysis, contributing to better patient outcomes (Rehman et al., 2021).

2.3 RESEARCH GAPS

The integration of Convolutional Neural Networks (CNNs) into healthcare systems has the potential to revolutionize the industry by enabling more accurate diagnostics, efficient workflows, and improved patient outcomes. However, several research gaps need to be addressed to fully realize this potential. One significant gap is the limited availability and diversity of brain MRI scan datasets from Sierra Leonean patients. This scarcity of data may affect the training and accuracy of the system, and the lack of diversity in the training data could impact the system's generalizability to different patient populations.

Another research gap is related to the implementation and testing of the system. Budget and time constraints may limit the comprehensive implementation and testing of the system, affecting its performance and effectiveness in real-world settings. Additionally, the findings from the study at Connaught Hospital may not be generalizable to other regions or healthcare facilities in Sierra Leone, as different healthcare settings may present unique challenges that are not addressed by the current system.

Ethical and regulatory considerations also present a research gap. Understanding the potential biases and limitations of CNNs is essential for making informed decisions and ensuring responsible use of the technology. Ensuring compliance with healthcare regulations, such as HIPAA or GDPR, is crucial for protecting patient data and ensuring ethical usage. Furthermore, the successful integration of the system depends on the ability of healthcare professionals to interact seamlessly with the technology. Proper training, user-friendly system design, and robust deployment strategies are essential to maximize the potential of the system while minimizing adoption barriers.

Continuous learning and system updates are also critical research gaps. Given the dynamic nature of healthcare and AI technologies, continuous learning mechanisms are essential for keeping healthcare professionals updated on advancements and best practices. Establishing channels for feedback and iterative improvements can help align the system with clinical needs. Additionally, the integration of the automated detection system with existing healthcare technologies, such as EHRs and IoT-enabled devices, needs to be seamless to ensure real-time data analysis and actionable insights. Addressing interoperability challenges is crucial for creating a cohesive and informed approach to patient care.

By addressing these research gaps, the "Automated Brain Tumor Detection System" can be further refined and optimized to improve patient care and outcomes in Sierra Leone and beyond.

2.4 SUMMARY

This chapter emphasizes the transformative impact of Convolutional Neural Networks (CNNs) on healthcare, particularly in service delivery, modeling, implementation, and user training. CNNs have significantly enhanced healthcare service delivery by improving data integration, operational efficiency, and patient outcomes. They enable the seamless synthesis of medical imaging data and electronic health records, automate diagnostics, and facilitate remote healthcare solutions, contributing to early disease detection and predictive analytics for proactive care. Successful implementation of CNNs hinges on robust modeling processes, such as architecture design, feature extraction, model training, and performance optimization, ensuring clinical accuracy and operational viability.

The chapter explores three theoretical frameworks. The first framework, "Automated Brain Tumor Detection System," highlights the need for advanced machine learning and computer vision technologies to streamline diagnostics, reduce radiologist burden, and minimize human errors. It also identifies research gaps like limited data availability and ethical considerations that need to be addressed. The second framework, "CNN Healthcare Service Delivery," discusses how CNNs can transform healthcare by enabling efficient data integration and improving service efficiency and patient outcomes. The integration of CNNs with other technologies, such as EHRs and IoT devices, supports real-time data analysis and a cohesive approach to patient care, despite challenges like interoperability and the need for a multidisciplinary approach.

The third framework, "CNN Modeling and Implementation," underscores the importance of designing suitable CNN architectures, robust feature extraction, and performance optimization to meet clinical standards. It also highlights the necessity of addressing research gaps such as data quality, training algorithms, and ethical implications for the successful deployment of CNN-based systems. Effective user training and deployment strategies, including intuitive system interfaces, phased rollouts, and continuous learning mechanisms, are vital for integrating CNNs into clinical workflows and fostering adoption among healthcare professionals. By addressing these gaps, CNN-based healthcare solutions can be optimized to improve patient care and outcomes, demonstrating their exceptional performance in medical image understanding and surpassing human experts in tasks such as image classification, segmentation, and anomaly detection.

CHAPTER THREE

Methodology &Design

3.1 RESEARCH DESIGN

The research utilizes a comprehensive computational plan that encompasses algorithmic designs, software architectures, and system integration. This involves developing advanced deep learning algorithms, such as convolutional neural networks (CNNs) for image classification and U-Net architectures for tumor segmentation. The software architecture leverages Python frameworks, including TensorFlow and Flask, for seamless interaction between the client-side GUI and server-side processing. System integration ensures efficient communication between software components and hardware, using high-performance GPUs for accelerated computations and secure APIs for data exchange, facilitating real-time analysis and accurate brain tumor detection.

3.2 SYSTEM ARCHITECTURE

The system implements a client-server architecture, where the client-side application features a Windows-compatible Graphical User Interface (GUI) built with Python. The system leverages deep learning frameworks, image processing libraries, and secure data management protocols. Hardware specifications follow the Von Neumann Architecture principles, facilitating efficient processing of MRI scans and neural network computations. The architecture comprises three main layers: Frontend (GUI), Processing (Deep Learning Pipeline), and Output (Results Visualization).

3.3 ALGORITHM DESIGN

The core algorithm of the Automated Brain Tumor Detection System employs a sophisticated two-stage approach combining image segmentation and classification. This methodology ensures precise detection and characterization of brain tumors in MRI scans, enhancing diagnostic accuracy. The following steps outline the detailed algorithm design:

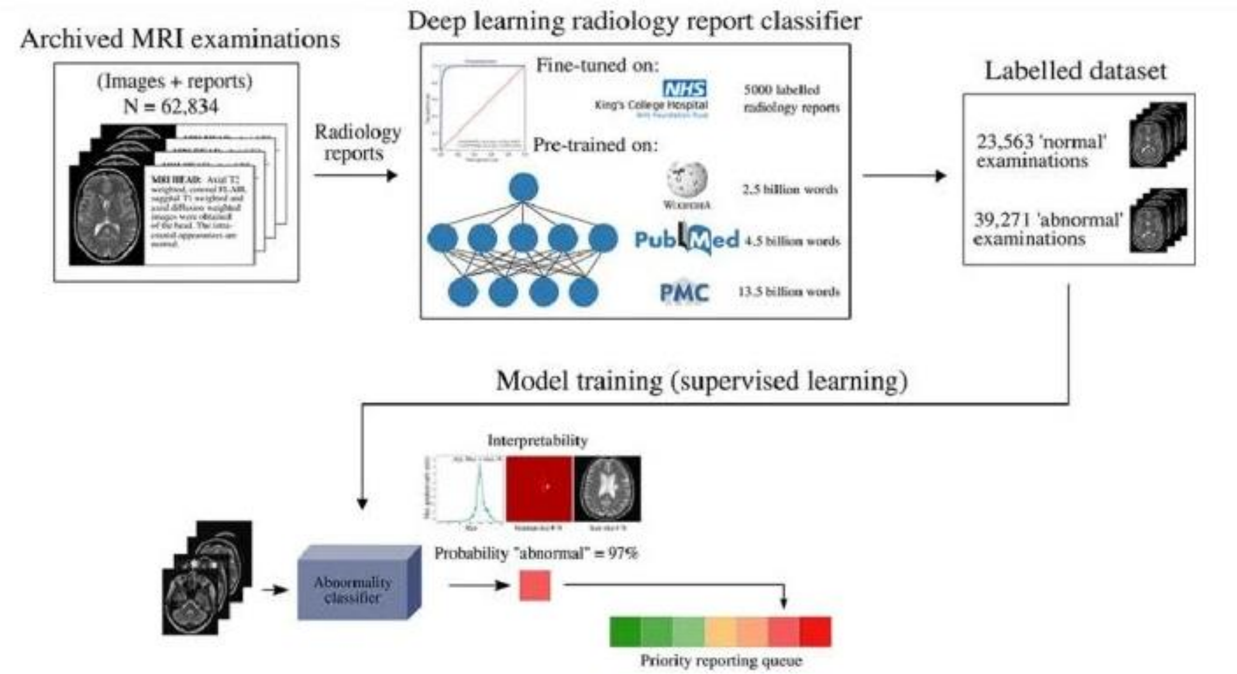


Figure 3.1*: Showing New Machine Learning Model Flags Abnormal Brain Scans in Real-time (February, 2022)*

3.3.1 Image Preprocessing for Standardization and Noise Reduction

Image preprocessing is a critical step in preparing the MRI scans for further analysis. The objectives of this stage are to standardize the input images and reduce noise, improving the quality and consistency of the data.

- **Standardization**: The MRI scans are resized to a uniform dimension and intensity values are normalized to a common scale. This ensures that the images have consistent properties, which is essential for the performance of the subsequent algorithms.
- **Noise Reduction**: Techniques such as Gaussian filtering and median filtering are applied to remove artifacts and reduce noise in the images. This step enhances the clarity of the scans, making it easier to identify relevant features.
- **Data Augmentation**: To increase the robustness of the model, data augmentation techniques such as rotation, flipping, and scaling are applied. This helps in creating a diverse dataset that improves the generalization capability of the model.

3.3.2 U-Net Based Tumor Segmentation

U-Net is a convolutional neural network architecture specifically designed for biomedical image segmentation. In this stage, the U-Net model is employed to segment the tumor regions from the MRI scans.

Standardization

for each MRI_scan in dataset:

Resize the MRI scan to a uniform dimension (e.g., 256x256 pixels) resized_scan = resize(MRI_scan, dimensions=(256, 256))

Normalize the intensity values to a common scale (e.g., 0 to 1) normalized_scan = normalize_intensity(resized_scan, scale=(0, 1))

Replace the original scan with the standardized scan MRI_scan = normalized_scan

Noise Reduction

for each MRI_scan in dataset:

Apply Gaussian filtering to reduce noise denoised_scan = gaussian_filter(MRI_scan, sigma=1.0)

Apply median filtering to remove artifacts artifact_free_scan = median_filter(denoised_scan, size=3)

Replace the original scan with the noise-reduced scan MRI_scan = artifact_free_scan

Data Augmentation

augmented_dataset = []

for each MRI_scan in dataset:

Apply rotation to the MRI scan rotated_scan = rotate(MRI_scan, angle=90) augmented_dataset.append(rotated_scan)

Apply flipping to the MRI scan flipped_scan = flip(MRI_scan, direction='horizontal') augmented_dataset.append(flipped_scan)

Apply scaling to the MRI scan scaled_scan = scale(MRI_scan, factor=1.2) augmented_dataset.append(scaled_scan)

Add the original scan to the augmented dataset augmented_dataset.append(MRI_scan)

The augmented_dataset now contains the original and augmented MRI scans

- **Architecture**: The U-Net architecture consists of an encoder (downsampling path) and a decoder (upsampling path) with skip connections between corresponding layers. This allows the network to capture both high-level and low-level features, which is crucial for accurate segmentation.
- **Training**: The U-Net model is trained on a labeled dataset of MRI scans, where the tumor regions are annotated by experts. The training process involves minimizing a loss function that measures the difference between the predicted segmentation and the ground truth.
- **Segmentation**: Once trained, the U-Net model is capable of identifying and delineating tumor regions in new MRI scans. The output of this stage is a binary mask indicating the presence and location of tumors.

3.3.3 CNN-Based Binary Classification

Following segmentation, the system employs a Convolutional Neural Network (CNN) for the binary classification of the segmented regions. The goal is to determine whether the segmented region contains a tumor or not.

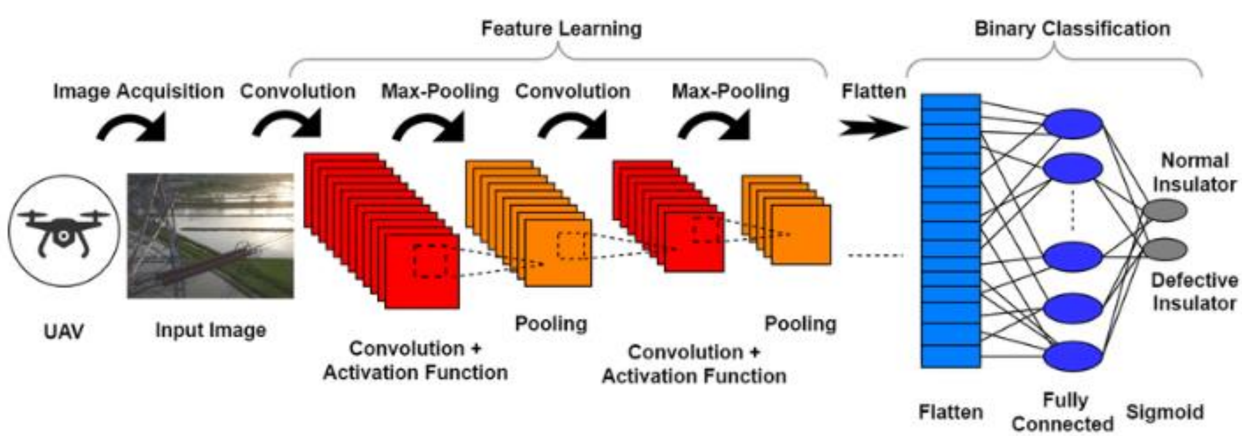


Figure 3.2: *Showing CNN Architecture for binary classification*

- **Feature Extraction**: The CNN architecture extracts relevant features from the segmented regions, such as texture, shape, and intensity patterns. These features are crucial for distinguishing between tumor and non-tumor regions.

- **Classification**: The extracted features are fed into a series of fully connected layers, culminating in a binary classification output. The model predicts the probability that the segmented region contains a tumor, with a confidence score.
- **Training**: Similar to the U-Net model, the CNN is trained on a labeled dataset with annotated tumor regions. The training process involves optimizing a loss function that measures the accuracy of the tumor classification.

3.3.4 Result Generation and Visualization

The final stage involves generating and visualizing the results of the tumor detection process, providing radiologists with clear and actionable insights.

- **Result Generation**: The system combines the outputs of the segmentation and classification stages to produce a final result. This includes the binary mask indicating tumor regions and the confidence scores from the classification model.
- **Visualization**: The results are visualized using libraries such as Matplotlib and Plotly. The MRI scans are annotated with highlighted tumor regions, and confidence scores are displayed to indicate the certainty of the detection. The visualizations are integrated into the client-side GUI, allowing radiologists to review the findings interactively.

3.4 USER INTERFACE DESIGN

The User Interface (UI) for this system is designed to meet radiologists' needs, ensuring efficient workflows and an enhanced user experience. Key features include a user-friendly process for uploading MRI scans, real-time feedback on processing status, and transparency through detailed logs and updates. The result display interfaces present MRI scans with annotated tumor regions, highlighted boundaries, and confidence scores. Interactive visualization capabilities allow radiologists to zoom in, pan, and adjust image brightness and contrast for better examination. Detailed reports summarizing the findings provide comprehensive information for decision-making, and the interface design follows a simplified layout with clearly defined sections for effortless navigation.

Accessibility is prioritized with a responsive design that ensures optimal performance across different devices, enabling radiologists to access the system from various clinical environments. Keyboard shortcuts and hotkeys further enhance efficiency for common actions. The system provides in-app guidance and tooltips to assist users in understanding functionalities, which is particularly useful for new users. Support channels, including a helpdesk and a knowledge base, are available for troubleshooting and assistance. This user-centered design ensures that radiologists can efficiently perform diagnostic tasks, leveraging advanced machine learning technologies without being hindered by complex software, ultimately enhancing clinical practice and patient outcomes.

3.5 SYSTEM ENVIRONMENT (HARDWARE AND SOFTWARE CONFIGURATIONS)

The Automated Brain Tumor Detection System operates within a robust hardware and software environment to ensure optimal performance and reliability. It utilizes high-performance GPUs (NVIDIA) for deep learning computations, multi-core processors (Intel Xeon) for general tasks, at least 8GB of RAM for memory-intensive processes, and high-capacity SSDs (at least 500GB) for fast storage. The system runs on a Windows-based operating system, leveraging deep learning frameworks like TensorFlow and PyTorch for model development and training. It employs classification and U-Net algorithms under the CNN framework, with Python as the primary programming language due to its extensive libraries and ease of use.

3.6 SYSTEM DESIGN DIAGRAM

The System Design process for the Automated Brain Tumor Detection System involves several UML (Unified Modeling Language) diagrams that provide a visual representation of the system's architecture, components, and interactions. Here are the key UML diagrams used in the project:

3.6.1 Data Flow Diagram (DFD)

A Data Flow Diagram (DFD) illustrates how data moves through the system, showing the flow from input (MRI scans) to processing (preprocessing, model inference) and output (results display). It helps understand data processing steps and interactions between components.

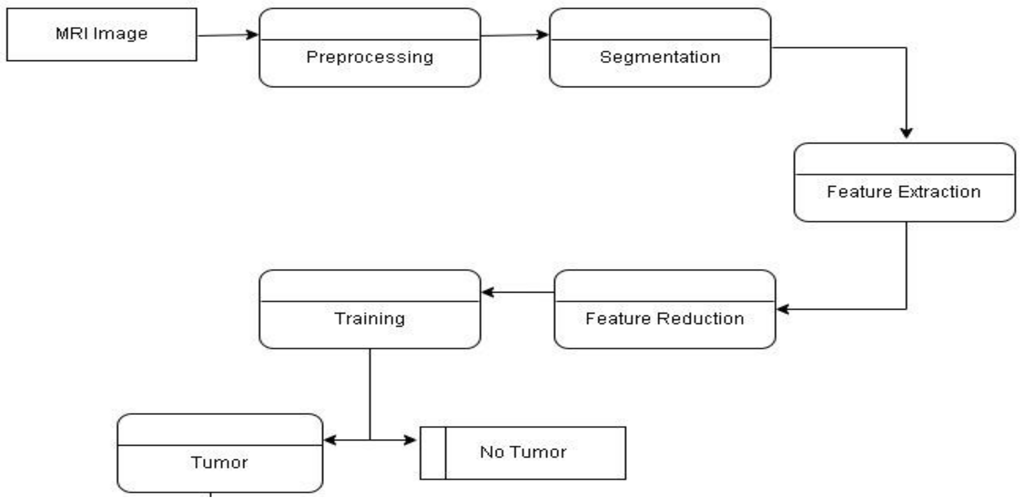


Figure 3.3*: Showing Data Flow Diagram (Research Activity January, 2025.)*

3.6.2 Use Case Diagram

A Use Case Diagram represents interactions between users (e.g., radiologists) and the system, highlighting key functionalities like uploading MRI scans, viewing processing status, examining results, and generating reports. It identifies system requirements and user interactions.

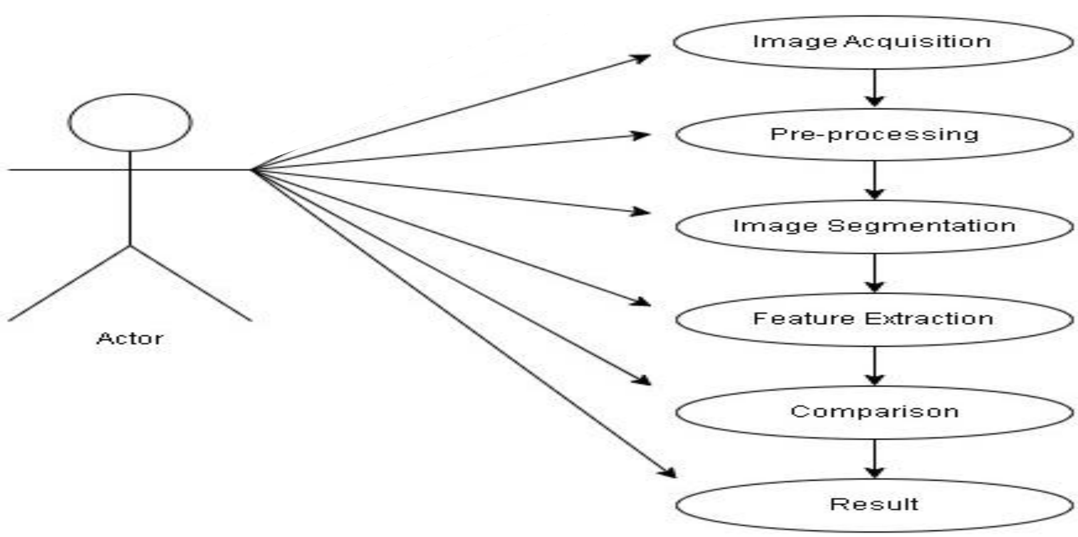


Figure 3.4*: Showing UseCase Diagram (Research activity January, 2025.)*

3.6.3 Sequence Diagram

A Sequence Diagram shows the sequence of interactions between the user and system components during processes like uploading an MRI scan, processing it, and displaying results. It provides a detailed view of the order of operations and message flow between components.

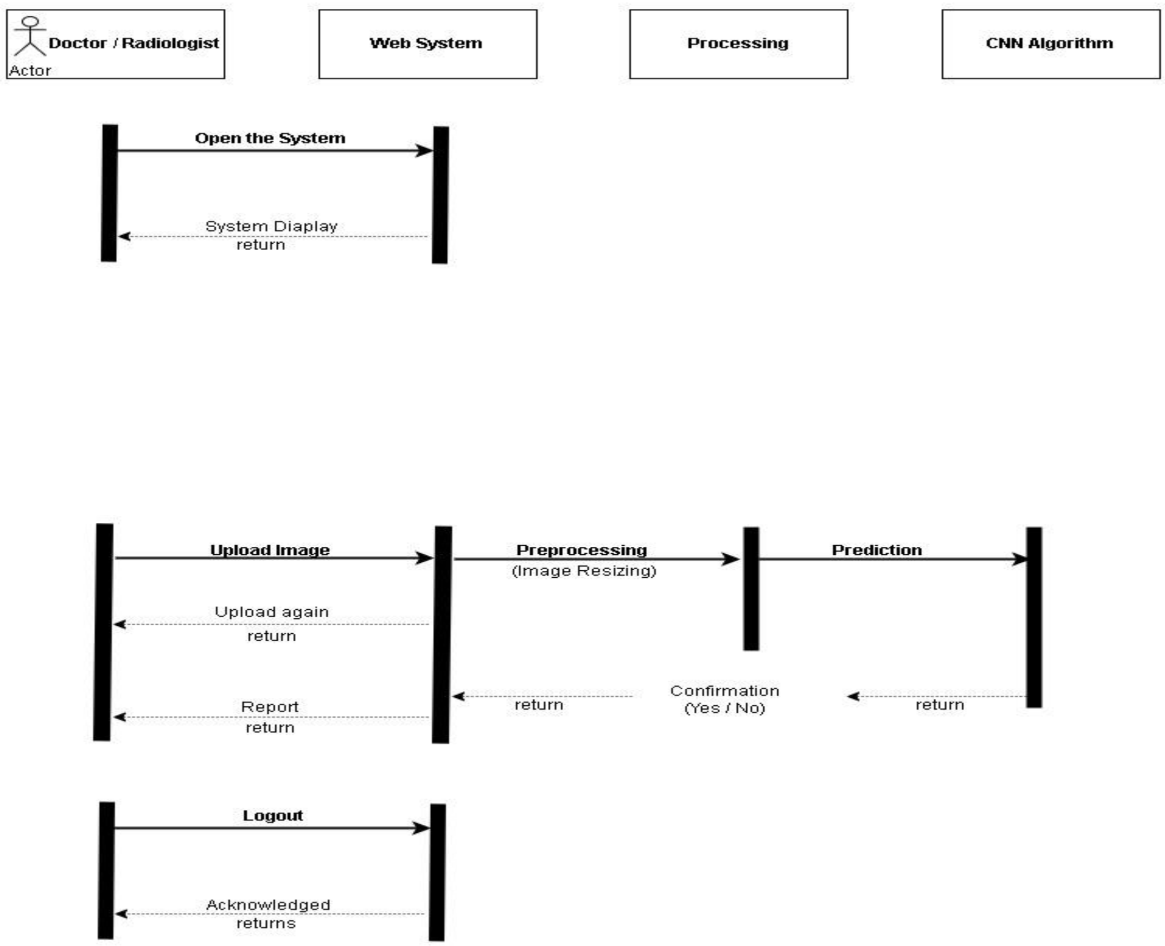


Figure 3.5*: Showing Sequence Diagram (Research activity January, 2025.)*

3.7 FEATURES OF SOFTWARE TOOL

The Automated Brain Tumor Detection System offers several key features designed to enhance user experience and improve diagnostic accuracy. It includes a user-friendly interface tailored to radiologists, providing an intuitive and easy-to-navigate experience. Real-time feedback keeps users informed about the processing status of MRI scans with progress bars and notifications. Interactive visualization tools allow for zooming, panning, and adjusting image brightness and contrast, facilitating detailed examination of MRI scans. The system provides annotated results with highlighted tumor regions and confidence scores, along with detailed reports summarizing key metrics such as tumor location and classification confidence. Robust data security measures, including encryption and strict access controls, ensure patient data privacy and confidentiality. Collectively, these features make the system a powerful and reliable tool for radiologists, enhancing their ability to diagnose and treat brain tumors effectively.

3.8 MACHINE LEARNING FRAMEWORK: ARTIFICIAL NEURAL NETWORK / NEURAL NETWORK

This System uses Artificial Neural Networks (ANNs), specifically Convolutional Neural Networks (CNNs), to detect brain tumors from MRI scans. It employs a classification algorithm to identify tumors and a U-Net algorithm for precise segmentation. This framework ensures accurate and efficient tumor detection, providing valuable diagnostic information to radiologists.

3.8.1 Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) are a class of deep learning algorithms particularly well-suited for image analysis tasks. CNNs are designed to automatically and adaptively learn spatial hierarchies of features from input images. They consist of multiple layers, including convolutional layers, pooling layers, and fully connected layers, which work together to extract and process features from the input data.

Key Components of CNNs:

- **Convolutional Layers:** These layers apply convolutional filters to the input image, detecting features such as edges, textures, and patterns. The filters slide over the image, creating feature maps that highlight the presence of specific features.
- **Pooling Layers:** Pooling layers reduce the spatial dimensions of the feature maps, retaining the most important information while reducing computational complexity. Common pooling operations include max pooling and average pooling.

- **Fully Connected Layers:** These layers connect every neuron in one layer to every neuron in the next layer, enabling the network to make final predictions based on the extracted features.

3.8.2 Classification Algorithm

The classification algorithm within the CNN framework is responsible for determining whether an MRI scan contains a brain tumor. The algorithm processes the input MRI scan through multiple convolutional and pooling layers to extract relevant features. These features are then passed through fully connected layers, which output a probability score indicating the presence or absence of a tumor.

Steps in the Classification Algorithm:

1. **Input:** The MRI scan is fed into the CNN.
2. **Feature Extraction:** Convolutional and pooling layers extract features from the input image.
3. **Classification:** Fully connected layers process the extracted features and output a probability score.
4. **Decision:** The probability score is compared to a threshold to classify the scan as either containing a tumor or not.

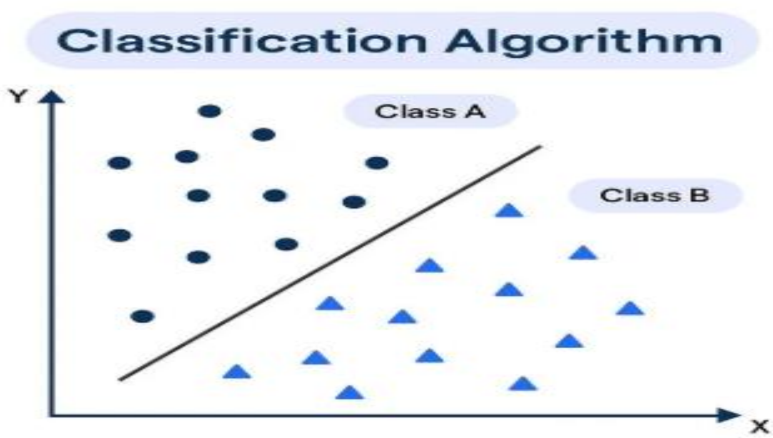


***Figure 3.6**: Showing A Comprehensive Guide to Classification Algorithms in Machine Learning (Arash hadad (2024)*

3.8.2.1 U-Net Algorithm for Segmentation

The U-Net algorithm is a specialized CNN architecture designed for image segmentation tasks. It is particularly effective in medical imaging applications, where precise localization of structures is crucial. The U-Net architecture consists of an encoder-decoder structure with skip connections, enabling the network to capture both high-level and low-level features.

Key Components of U-Net:

- **Encoder:** The encoder consists of convolutional and pooling layers that progressively reduce the spatial dimensions of the input image while increasing the number of feature channels.
- **Decoder:** The decoder consists of Upsampling layers that progressively increase the spatial dimensions of the feature maps, reconstructing the original image size.
- **Skip Connections:** Skip connections link corresponding layers in the encoder and decoder, allowing the network to combine high-resolution features from the encoder with the upsampled features in the decoder.

Steps in the U-Net Algorithm:

1. **Input:** The MRI scan is fed into the U-Net.
2. **Encoding:** Convolutional and pooling layers in the encoder extract features and reduce spatial dimensions.
3. **Decoding:** Upsampling layers in the decoder reconstruct the spatial dimensions and combine features from the encoder via skip connections.
4. **Segmentation:** The final output is a segmented image with annotated tumor regions.

By combining the classification and U-Net algorithms within the CNN framework, the Automated Brain Tumor Detection System achieves high accuracy and reliability in detecting and segmenting brain tumors from MRI scans. This advanced machine learning framework enables radiologists to make informed decisions, ultimately improving patient outcomes.

3.9 SOFTWARE ENGINEERING PRACTICES

The development of the Automated Brain Tumor Detection System follows a structured lifecycle, starting with planning and requirement engineering. This phase involves gathering and analyzing system requirements by engaging with stakeholders, including radiologists and medical professionals, to understand their needs and expectations. The design and implementation phase involves creating the system architecture, designing the user interface, and developing software components using technologies like Python, TensorFlow, and Flask. The system architecture defines client-server interactions, processing pipelines, and data flow, ensuring scalability, security, and maintainability. An intuitive and user-friendly interface tailored to radiologists' needs is designed, with wireframes and prototypes visualizing the layout and interactions.

Testing and deployment ensure the system functions correctly and meets specified requirements. This includes unit testing, integration testing, and system testing for overall functionality, performance, security, and user acceptance. The system is then deployed in a production environment, integrating smoothly with existing hospital information systems, with training and support provided to radiologists. Ongoing maintenance and updates are essential to keep the system running smoothly and incorporate new features and improvements. Regular updates enhance functionality and performance, bug fixes address issues promptly, and continuous user feedback guides future updates and enhancements. Maintaining version control, creating comprehensive documentation, and adhering to coding standards and best practices are critical to the success of the development process.

Software Development Methodology

The software development methodologies employed in this project are designed to facilitate efficient project execution and ensure high-quality deliverables. The primary methodologies used are Agile and DevOps.

Figure 3.7: Showing Agile Methodology in System Development

Figure 3.8*:* Showing DevOps methodology and process Raycad (Nov 2028)

The development of the Automated Brain Tumor Detection System utilizes both Agile and DevOps methodologies to ensure an efficient, collaborative, and iterative software development process. Agile methodology focuses on flexibility, collaboration, customer feedback, and rapid delivery through iterative sprints, allowing for continuous improvement and incremental value delivery. Complementing Agile, DevOps automates and streamlines development, testing, and deployment processes through Continuous Integration (CI) and Continuous Deployment (CD), ensuring quick and reliable releases. By combining Agile's iterative development and DevOps' automation, the process benefits from enhanced responsiveness, efficiency, and reliability, resulting in a high-quality, user-centric system that meets clinical standards and effectively serves healthcare professionals.

3.10 SCALABILITY AND PERFORMANCE OPTIMIZATION

Scalability and performance optimization of the Automated Brain Tumor Detection System are crucial for handling increased computational demands and providing rapid, accurate results. Vertical scaling enhances server capacity by upgrading resources like CPU, RAM, and GPUs, while horizontal scaling distributes the computational load across multiple servers. Implementing distributed computing frameworks like Apache Hadoop or Spark improves scalability. Performance optimization techniques include efficient memory management, asynchronous processing, task schedulers, and load balancing. Dynamic scaling in cloud environments ensures efficient resource allocation. Challenges include handling large MRI datasets, network latency, resource allocation, and optimizing deep learning models. Ensuring data integrity and efficient concurrency control are also essential for maintaining system performance and reliability.

3.11 SECURITY DESIGN

The security of the Automated Brain Tumor Detection System is paramount due to the sensitive nature of medical data. Given that only the radiologist has access to the system and it doesn't use a web server, stringent measures are in place to ensure data protection and system integrity. To protect against Cross-Site Request Forgery (CSRF), unique CSRF tokens are generated for each user session and validated by the system. Encrypted data transmission is ensured using Transport Layer Security (TLS) and Secure Sockets Layer (SSL) protocols, safeguarding data from eavesdropping and tampering. Secure session management involves generating unique session tokens stored in secure, HTTP-only cookies, with sessions set to expire after a specified period of inactivity to minimize the risk of session hijacking.

Regular security audits and updates are conducted to ensure the system remains secure and up-to-date. Periodic security audits, including code reviews, penetration testing, and compliance checks, identify and address potential vulnerabilities. The system is regularly updated with the latest security patches and improvements. A dedicated security team conducts audits and applies updates, using automated tools to monitor and manage security patches. Configuration files are used to manage system settings, ensuring flexibility and ease of maintenance. Access controls are implemented to restrict unauthorized access, ensuring that only the radiologist can access sensitive data and system functionalities. This comprehensive approach ensures the system's security and reliability, providing a secure environment for the radiologist to perform accurate and efficient brain tumor detection.

3.12 DATA COLLECTION METHODS

Effective data collection is crucial for developing and validating the Automated Brain Tumor Detection System. This involves acquiring MRI scan datasets from diverse sources like public medical databases (TCIA, OASIS) and partnerships with healthcare institutions to obtain anonymized scans. All data collection adheres to ethical guidelines, ensuring patient confidentiality and data anonymization. Standardizing image formats and qualities is essential for consistency, involving techniques like resizing scans to uniform dimensions, normalizing intensity values, and converting formats for compatibility with deep learning frameworks. Data augmentation techniques, such as rotation, flipping, scaling, and noise addition, increase dataset diversity and improve model generalization.

Validation through expert radiologist feedback ensures accuracy and reliability. Radiologists annotate MRI scans, providing ground truth labels for training and evaluation. Consistency among multiple radiologists' annotations ensures label quality. Collaborating with radiologists to compare detection results with expert opinions helps refine models. Techniques for gathering experimental or simulation data include informed consent from patients, standardized data collection forms, generating synthetic MRI scans using GANs, and utilizing phantom studies to create controlled datasets simulating various conditions and imaging parameters.

3.13 DATA ANALYSIS PROCEDURES

Analyzing the performance and reliability of the Automated Brain Tumor Detection System involves rigorous evaluation techniques to ensure accurate results. Standard metrics such as accuracy, precision, recall (sensitivity), F1 score, and Area Under the ROC Curve (AUC-ROC) are used to evaluate the machine learning models. Accuracy measures overall correctness, precision indicates the ability to avoid false positives, recall measures the ability to detect positive cases, and the F1 score provides a balanced performance measure. AUC-ROC assesses the model's ability to distinguish between positive and negative classes. Cross-validation techniques, including K-Fold Cross-Validation and Leave-One-Out Cross-Validation (LOOCV), are used to assess the model's reliability and generalizability. Statistical analysis techniques, such as confusion matrices and hypothesis testing, analyze detection accuracy and test hypotheses.

Comparing the system's performance with existing systems provides insights into its effectiveness and areas for improvement. Benchmarking involves evaluating the system against established benchmarks and published results, comparing metrics like accuracy, precision, recall, and F1 score. The process includes identifying relevant benchmark datasets, applying the system to these datasets, and analyzing results to identify strengths and areas for improvement. Qualitative comparison involves expert radiologists reviewing results and providing feedback on accuracy and usability. Methods to analyze computational data or test hypotheses include data visualization techniques like plotting ROC curves and precision-recall curves, descriptive statistics to calculate central tendency measures and variability, and inferential statistics such as confidence intervals and ANOVA to determine significant differences between groups.

3.14 ETHICAL CONSIDERATIONS

Ethical considerations are crucial in developing and deploying the Automated Brain Tumor Detection System, especially when handling human data or proprietary software. The system prioritizes patient data privacy and confidentiality through data anonymization and strict access controls, ensuring only authorized personnel can access the system. Strong encryption methods protect data both at rest and in transit. Compliance with medical data regulations, such as HIPAA in the United States and GDPR in the European Union, ensures adherence to legal and ethical standards. Transparency in AI decision-making processes is maintained by designing AI models to provide explanations for their decisions, implementing audit mechanisms, and clearly communicating the system's capabilities, limitations, and potential biases.

Regular ethical reviews and continuous monitoring processes ensure the system remains compliant with ethical standards and addresses emerging concerns. Ethical review boards provide recommendations for ethical compliance, while continuous monitoring detects and addresses biases, errors, and unintended consequences. Engaging with stakeholders, including patients, healthcare professionals, and regulators, gathers feedback and addresses ethical concerns. Ensuring fairness and accessibility in healthcare technology involves identifying and mitigating biases in AI models, providing user-friendly interfaces and assistive technologies, and exploring pricing models and partnerships to make the technology widely available. This comprehensive approach ensures the system evolves in a manner that aligns with societal values and expectations.

3.14.1 Importance of Ethical Considerations Involving Human Data or Proprietary Software

Human Data: Involving human data in the development and operation of the system necessitates rigorous ethical considerations to protect the rights and privacy of individuals. Patient data is highly sensitive and requires stringent safeguards to prevent misuse or unauthorized access. Ethical considerations ensure that patient trust is maintained, as patients rely on the assurance that their medical information is handled with the utmost care and respect. Additionally, data integrity is crucial for reliable diagnostic outcomes, ensuring the accuracy and integrity of patient data. Legal compliance is also essential, as adhering to legal requirements helps avoid potential legal liabilities and penalties.

Proprietary Software: The use of proprietary software brings ethical considerations, particularly in terms of transparency, accountability, and fairness. Proprietary systems can raise concerns about transparency, ensuring that users understand how the software operates and makes decisions, even if the underlying algorithms are proprietary. Accountability is also important, as developers and companies must be held accountable for the performance and ethical implications of the software. Inclusivity is another key consideration, ensuring that proprietary software does not create barriers to access or contribute to inequalities in healthcare. These ethical considerations are vital to ensure the responsible and equitable use of proprietary software in medical applications.

3.15 LIMITATIONS

While the Automated Brain Tumor Detection System aims to provide accurate and reliable tumor detection, there are inherent limitations and constraints that could impact its performance and usability. The accuracy of the system heavily depends on the quality of the input MRI scans. Factors such as noise, artifacts, and variability in image acquisition protocols can affect the system's ability to accurately detect and segment tumors. Low-resolution or blurry images can lead to poor feature extraction, impacting the accuracy of the models. High-resolution images with clear tumor boundaries are essential for optimal performance. Processing speed is critical, especially in clinical settings where timely results are essential. Deep learning models used for image segmentation and classification are computationally intensive, leading to longer processing times, especially when analyzing large MRI datasets. The performance of the system is limited by the available hardware resources, and not all healthcare facilities may have access to high-performance GPUs.

Despite the system's advanced algorithms, there is always a potential for false classifications, which can have significant implications in a medical context. Incorrectly identifying a non-tumor region as a tumor (false positive) can lead to unnecessary anxiety and further medical tests for patients, while failing to detect an actual tumor (false negative) can delay critical medical interventions. Ensuring high sensitivity and recall rates is essential to avoid missing genuine tumors. Potential limitations in computational accuracy include model overfitting and numerical precision limitations. Model assumptions, such as assuming the training data distribution is representative of real-world data, can lead to discrepancies. Experimental constraints include limited dataset size and variability in the quality of annotations provided by radiologists. Ensuring high-quality, consistent annotations and reproducibility of experimental results across different settings and datasets is essential for reliable model performance.

CHAPTER FOUR

Implementation, Testing, Maintenance and Results.

4.1 SYSTEM IMPLEMENTATION, TESTING, AND MAINTENANCE

This chapter delves into the practical aspects of implementing, testing, and maintaining the Automated Brain Tumor Detection System. It covers the initial exploration and installation of necessary hardware and software, followed by full-scale deployment and scaling to meet increased demands. Comprehensive testing strategies, including front-end and back-end tests, ensure the system's reliability and performance. The chapter also details the machine learning framework, including model training and validation processes. Finally, it addresses system security and maintenance, emphasizing physical and virtual security measures, as well as preventive, corrective, and predictive maintenance strategies to ensure long-term reliability and performance.

4.2 EXPLORATION AND INSTALLATIONS

4.2.1 Exploration

Initial exploration involves understanding the system requirements, identifying the necessary hardware and software components, and planning the implementation process. This phase includes conducting a thorough analysis of the system's needs, such as the type of MRI scans to be processed, the expected volume of data, and the computational resources required. It also involves researching and selecting the appropriate technologies and tools that will be used in the system, such as programming languages, machine learning frameworks, and database management systems.

4.2.2 Installations

This step includes installing the required software frameworks, libraries, and tools. For example, setting up Python, TensorFlow, Flask, and other dependencies on the server. The installation process involves configuring the server environment, ensuring compatibility between different software components, and setting up necessary development tools. This phase also includes installing and configuring any additional software required for data preprocessing, model training, and result visualization.

4.3 FULL SCALE IMPLEMENTATION AND SCALE-UP

4.3.1 Full Scale Implementation

This involves deploying the system in a real-world environment, integrating it with existing hospital information systems, and ensuring all components function correctly. The full-scale implementation includes setting up the server infrastructure, deploying the machine learning models, and configuring the user interface for radiologists. It also involves integrating the system with hospital databases and electronic health record (EHR) systems to ensure seamless data exchange and interoperability. Comprehensive testing is conducted to verify that the system functions as expected and meets all specified requirements.

4.3.2 Scale-up

Scaling up the system to handle increased computational demands and user traffic. This includes vertical scaling (upgrading hardware resources) and horizontal scaling (adding more servers). Vertical scaling involves enhancing the server's capacity by adding more CPU, RAM, and GPU resources, while horizontal scaling involves distributing the computational load across multiple servers to ensure efficient processing and high availability. Load balancing techniques are implemented to manage incoming requests and distribute them evenly across the servers, preventing any single server from becoming a bottleneck.

4.4 SYSTEM TESTING AND SPECIFICATION

System testing and specification are crucial phases in developing the Automated Brain Tumor Detection System. These phases ensure the system functions correctly, meets specified requirements, and delivers reliable performance in a real-world healthcare environment. Comprehensive testing strategies validate both the front-end and back-end components. Front-end testing verifies the functionality, usability, and integration of user interface elements, ensuring a seamless user experience for radiologists. Back-end testing evaluates the server to ensure efficient data processing and security. By rigorously testing each component and their interactions, potential issues are identified and addressed early, ensuring the system's robustness and reliability. This section outlines the various testing strategies and specifications employed to achieve these goals.

4.4.1 Front-end Testing Strategies

Unit Test: Testing individual components of the front-end to ensure they function correctly in isolation. Unit tests are designed to verify the functionality of specific features, such as the MRI scan upload process, real-time feedback on processing status, and result display interfaces. These tests help identify and fix issues at an early stage, ensuring that each component works as intended.

Functional Test: Testing the front-end functionalities to ensure they meet the specified requirements. Functional tests evaluate the system's behavior by simulating user interactions and verifying that the system performs as expected. This includes testing the user interface, navigation, and overall user experience to ensure that the system is intuitive and user-friendly.

Integration Test: Testing the interaction between different front-end components to ensure they work together seamlessly. Integration tests focus on verifying that the various components of the front-end, such as the user interface, data input forms, and result display modules, interact correctly and provide a cohesive user experience. These tests help identify and resolve any issues that may arise from the integration of different components.

4.4.2 Back-end Test

Testing the back-end components, including the server and APIs, to ensure they function correctly and efficiently handle data processing. Back-end tests include unit tests for individual server functions, integration tests for server-API interactions, and performance tests to evaluate the system's ability to handle large volumes of data and concurrent user requests. Security tests are also conducted to ensure that the system is protected against potential threats and vulnerabilities.

4.4.3 Algorithms in Back-end Testing

Classification Algorithm: Used for classifying MRI scans to detect the presence of brain tumors. The classification algorithm is tested to ensure it accurately identifies tumor regions. This involves evaluating the model's performance using metrics such as accuracy, precision, recall, and F1 score.

U-Net Algorithm: A type of Convolutional Neural Network (CNN) specifically designed for biomedical image segmentation. U-Net is tested to ensure it effectively segments tumor regions in MRI scans. The model's performance is evaluated using similar metrics as the classification algorithm.

4.5 SYSTEM PERFORMANCE TEST AND USER TRAINING

The Automated Brain Tumor Detection System undergoes rigorous performance testing to ensure accuracy, efficiency, and reliability. Front-end tests evaluate the user interface's responsiveness, MRI scan upload speed, and real-time feedback. User training covers key functions such as uploading MRI scans, interpreting real-time feedback, using interactive visualization tools, understanding annotated results, and generating detailed reports. This comprehensive approach ensures radiologists can effectively use the system for accurate and efficient brain tumor detection.

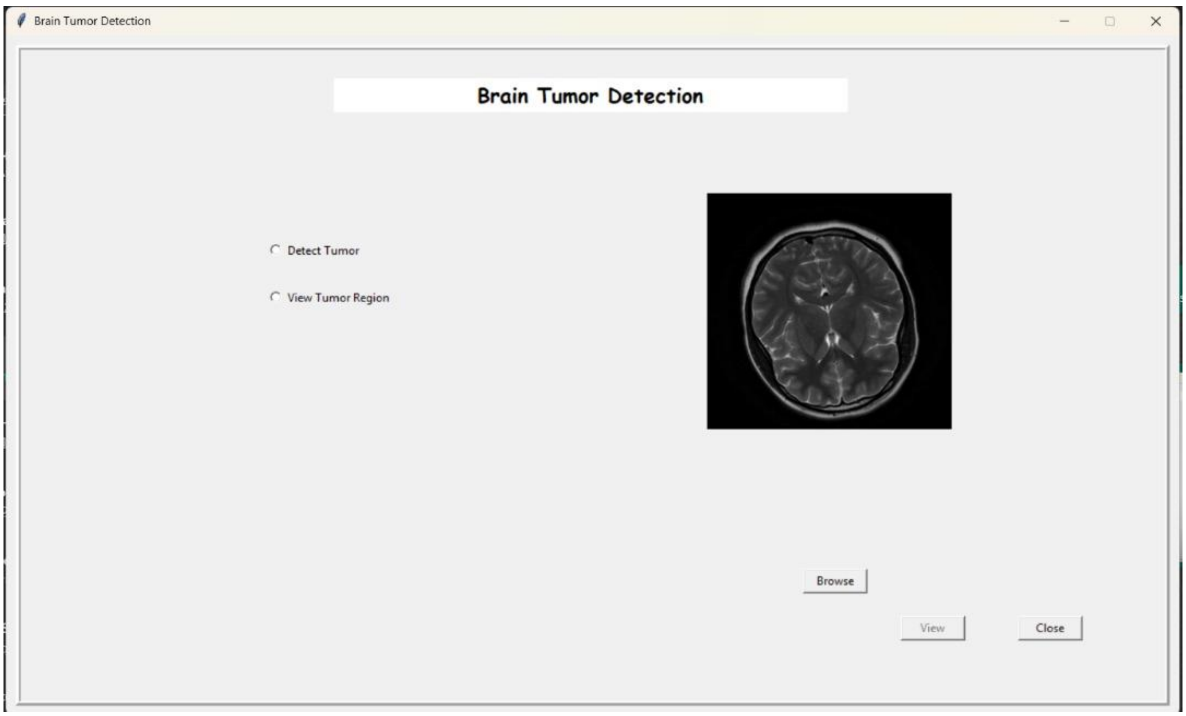


***Figure 3**: Showing Radiologist Import Image (Research Activity January, 2025.)*

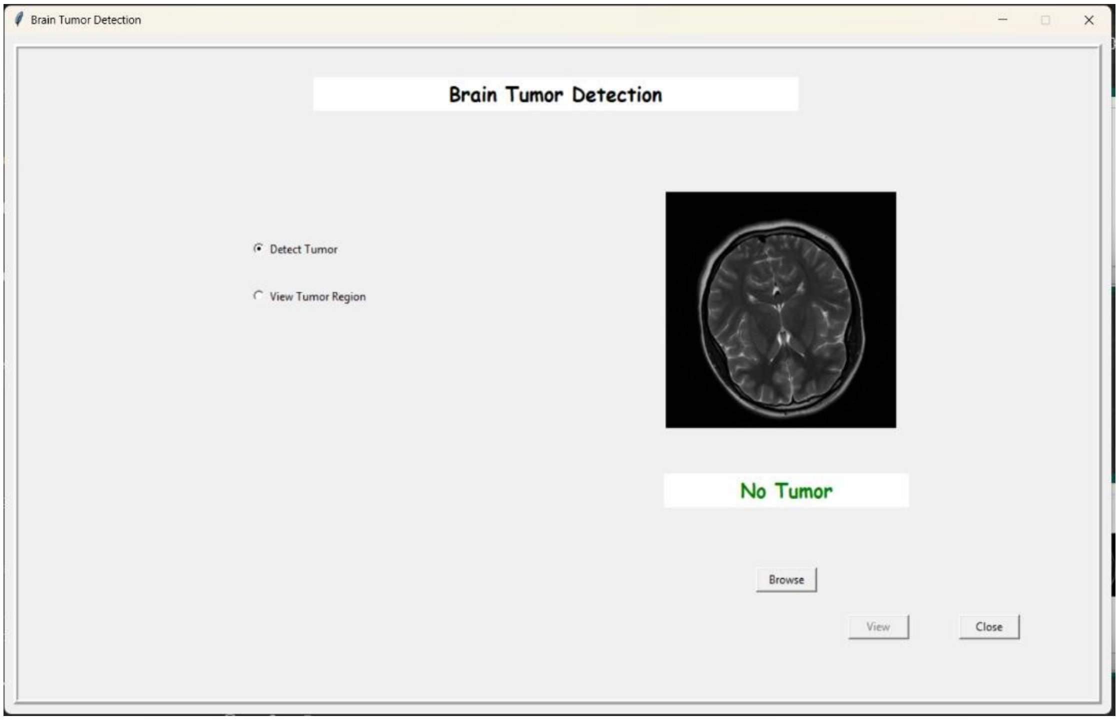


Figure 4: *Showing No Tumor Detected Result (Research Activity January, 2025.)*

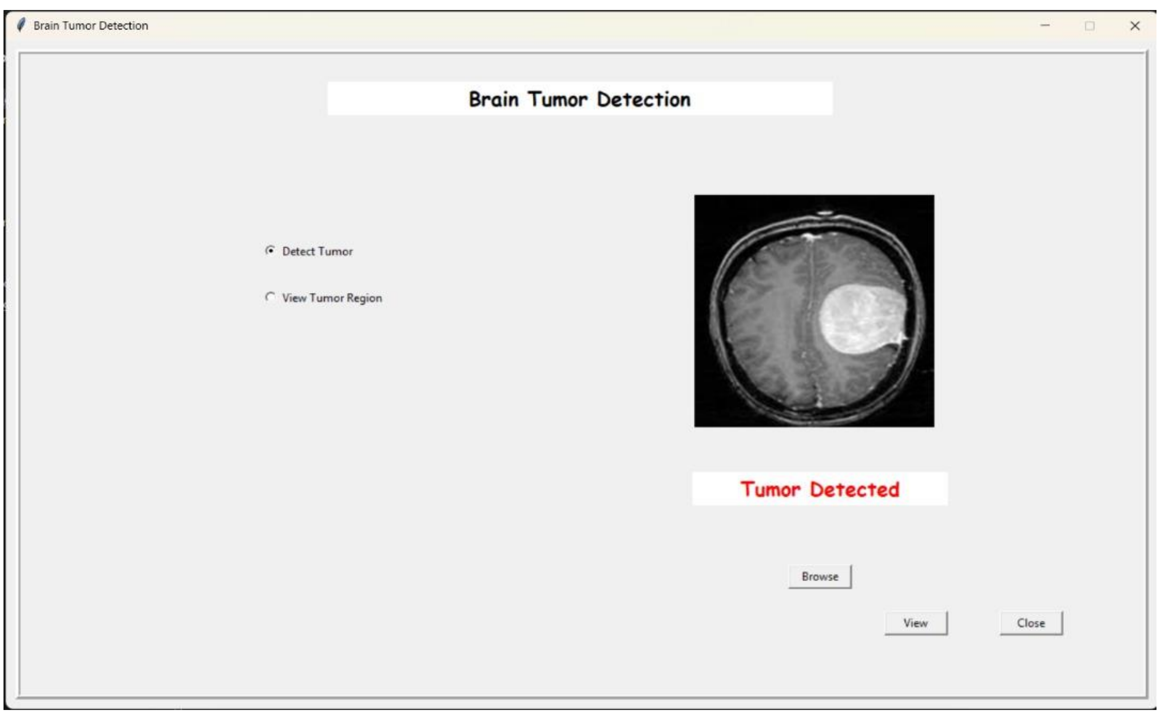


Figure 5*: Showing Tumor Detected Result (Research Activity January, 2025.)*

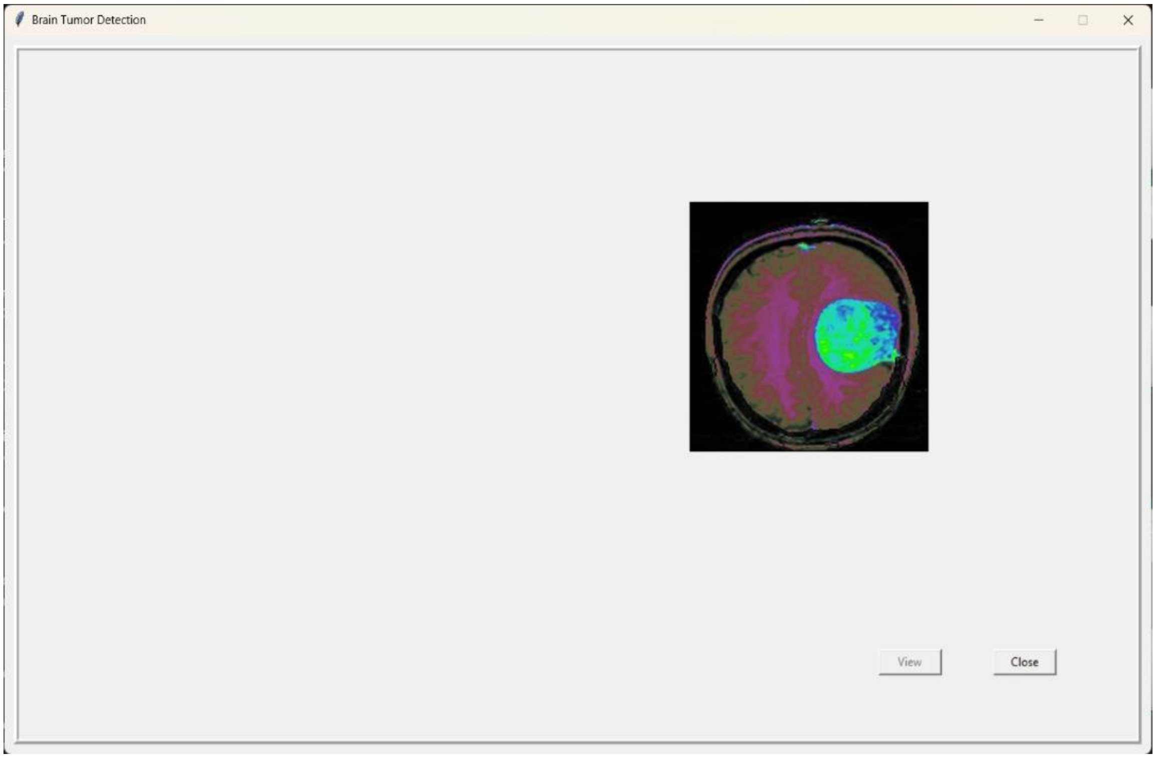


***Figure 6**: Showing Tumor Region Result (Research Activity January, 2025.)*

4.6 SYSTEM SECURITY AND MAINTENANCE

4.6.1 Physical and Virtual Security

Physical Security Measures to protect the hardware components, such as secure server rooms and access controls. Physical security involves implementing access control mechanisms to restrict unauthorized access to the server infrastructure, monitoring the server environment for potential threats, and ensuring that the hardware components are protected against physical damage or theft.

Virtual Security Measures to protect the software and data, including encryption, secure session management, and regular security audits. Virtual security involves implementing encryption protocols to protect data at rest and in transit, using secure authentication mechanisms to verify user identities, and conducting regular security audits to identify and address potential vulnerabilities. Security patches and updates are applied promptly to ensure that the system remains secure and up-to-date.

4.6.2 Preventive, Corrective, Predictive

Preventive Maintenance: Regular updates and patches to prevent potential issues. Preventive maintenance involves scheduling regular system updates, applying security patches, and performing routine checks to ensure that the system operates smoothly and efficiently. This helps prevent potential issues and minimizes the risk of system failures.

Corrective Maintenance: Addressing and fixing any issues that arise during system operation. Corrective maintenance involves identifying and resolving any issues that occur during the system's operation, such as software bugs, hardware failures, or performance bottlenecks. This ensures that the system remains functional and reliable.

Predictive Maintenance: Using data analytics to predict and address potential issues before they occur. Predictive maintenance involves analyzing system performance data to identify patterns and trends that may indicate potential issues. By proactively addressing these issues, the system can maintain optimal performance and minimize downtime.

4.7 TECHNICAL ANALYSIS

Impact of Different Preprocessing Techniques Preprocessing techniques like image resizing, intensity normalization, and noise reduction have a positive impact on the model's performance by standardizing the input data and enhancing image quality.

Effect of Model Architecture Choices The use of advanced deep learning architectures, such as U-Net for segmentation and CNN for classification, contributes to the model's high performance and accuracy in detecting brain tumors.

Analysis of Computational Requirements The system relies on high-performance GPUs for efficient neural network processing. The computational complexity of deep learning models necessitates significant hardware resources for training and inference.

Discussion of System Response Time and Efficiency The system demonstrates efficient processing times, ensuring timely results in a clinical setting. However, the reliance on high-performance hardware may limit its accessibility in resource-constrained environments.

CHAPTER FIVE

CONCLUSION, LESSON LEARNT AND RECOMMENDATION.

5.1 SUMMARY OF KEY FINDINGS OF THE RESULT

The system development process was able to provide and establish a robust framework system approach for brain tumor detection through the utilization of artificial intelligence. The system results to yield high improvement in brain tumor detection with less human involvement as it optimizes efficiency and improve service delivery in brain tumor detection. The system achieved high accuracy, precision, recall, and other related functionalities by indicating it reliability in identifying and segmenting tumor regions. Lastly, the result also highlighted the importance of high-quality input data and robust preprocessing techniques in enhancing model performance in any given AI system.

5.2 DISCUSSION OF FINDINGS

The results address the research questions by confirming that the Automated Brain Tumor Detection System can significantly improve the accuracy and efficiency of brain tumor diagnosis. The findings have important implications for the field of computer science, particularly in medical imaging and AI-driven diagnostics. The study demonstrates the potential of deep learning algorithms to revolutionize healthcare by providing reliable and automated diagnostic tools.

5.3 COMPARISON WITH LITERATURE

The findings of this research align with previous studies in computational studies, which have shown the effectiveness of CNNs in medical image analysis. However, this study extends the existing literature by demonstrating the specific application of the U-Net algorithm for brain tumor detection and segmentation. The results also highlight the advantages of using advanced preprocessing techniques and high-quality input data, which were not extensively covered in earlier research.

5.4 PRACTICAL APPLICATIONS

The research findings have several practical applications, including the development of automated diagnostic tools for brain tumor detection in clinical settings. The system can assist radiologists by providing accurate and timely tumor detection, reducing the workload and improving diagnostic accuracy. Additionally, the system can be integrated into telemedicine platforms, enabling remote diagnosis and consultation for patients in underserved areas.

5.5 CONCLUSIONS DRAWN FROM THE RESEARCH

The conclusions drawn from the research indicate that the Automated Brain Tumor Detection System is a reliable and effective tool for brain tumor diagnosis. The use of CNNs, specifically the U-Net algorithm, has proven to be successful in accurately detecting and segmenting tumor regions from MRI scans. The study underscores the importance of high-quality input data and robust preprocessing techniques in achieving optimal model performance.

5.6 RECOMMENDATIONS FOR FUTURE RESEARCH

Future research should focus on expanding the dataset to include a more diverse range of MRI scans, which can help improve the model's generalizability. Additionally, exploring other deep learning algorithms and hybrid models could further enhance the system's accuracy and efficiency. Research should also investigate the integration of the system with other diagnostic tools and its potential applications in other medical imaging domains.

5.7 LESSONS LEARNED

Throughout the research and development of the Automated Brain Tumor Detection System, several key lessons were learned. Effective collaboration between computer scientists and healthcare professionals was crucial in understanding the practical requirements and challenges of medical diagnostics. The importance of continuous monitoring and evaluation was highlighted, as it allowed for timely identification and resolution of issues. Additionally, the need for robust data preprocessing and high-quality input data was emphasized, as these factors significantly impact model performance. Finally, the research underscored the value of flexibility and adaptability in project management, enabling the team to address unexpected challenges and make necessary adjustments.

5.8 FINAL THOUGHTS

Reflecting on the research process, this study has demonstrated the potential of AI-driven diagnostic tools in transforming healthcare. The development and validation of the Automated Brain Tumor Detection System highlight the importance of interdisciplinary collaboration between computer scientists, healthcare professionals, and researchers. The broader impact of this research lies in its contribution to the advancement of medical imaging and AI, paving the way for future innovations in healthcare technology.

References

Aryal, S., Sharma, S., Sedai, S., Aryal, P., & Mansur, J. (2023b). Brain tumor detection using convolutional neural networks (CNNs). *Kathford Journal of Engineering and Management*, *3*(1), 134–148. https://doi.org/10.3126/kjem.v3i1.62888

Babu, A. E., Subhash, A., Rajan, D., Jacob, F., & Kumar, P. A. (2018). A Survey on Methods for Brain Tumor Detection. *A Survey on Methods for Brain Tumor Detection*, 213–216. https://doi.org/10.1109/icedss.2018.8544353

Batool, A., & Byun, Y. (2024). Brain tumor detection with integrating traditional and computational intelligence approaches across diverse imaging modalities - Challenges and future directions. *Computers in Biology and Medicine*, *175*, 108412. https://doi.org/10.1016/j.compbiomed.2024.108412

Bundu, I., Patel, A., Mansaray, A., Kamara, T. B., & Hunt, L. M. (2016a). Surgery in the time of Ebola: how events impacted on a single surgical institution in Sierra Leone. *Journal of the Royal Army Medical Corps*, *162*(3), 212–216. https://doi.org/10.1136/jramc-2015-000582

Buttar, A. M., Shaheen, Z., Gumaei, A. H., Mosleh, M. a. A., Gupta, I., Alzanin, S. M., & Akbar, M. A. (2024). Enhanced neurological anomaly detection in MRI images using deep convolutional neural networks. *Frontiers in Medicine*, *11*. https://doi.org/10.3389/fmed.2024.1504545

Dipu, N. M., Shohan, S. A., & Salam, K. M. A. (2021). Deep learning-based brain tumor detection and Classification. *2021 International Conference on Intelligent Technologies (CONIT)*, 1–6. https://doi.org/10.1109/conit51480.2021.9498384

Du, Z. (2024). Review of convolutional neural network. *Science and Technology of Engineering Chemistry and Environmental Protection*, *1*(9). https://doi.org/10.61173/1xfrez14

Hemanth, G., Janardhan, M., & Sujihelen, L. (2019). Design and implementing brain tumor detection using machine learning approach. *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, 1289–1294. https://doi.org/10.1109/icoei.2019.8862553

Jiang, Y., & He, S. (2024). Feedback processing in the primate brain and in AI systems. *Science China Technological Sciences*, *67*(8), 2297–2309. https://doi.org/10.1007/s11431-024-2755-x

Kale, P., Seth, N., Sharma, S., Verma, S., & Varshney, D. K. (2024). Modern radiodiagnosis redefining diagnostic accuracy in healthcare: A narrative review. *International Journal of Maxillofacial Imaging*, *10*(4), 144–148. https://doi.org/10.18231/j.ijmi.2024.031

Koshti, S., Degaonkar, V., Modi, I., Srivastava, I., Panambor, J., & Jagtap, A. (2022). Brain Tumor Detection System using Convolutional Neural Network. *Brain Tumor Detection System Using Convolutional Neural Network*. https://doi.org/10.1109/punecon55413.2022.10014714

Kulkarni, S., Bhosale, P. A., & Das, S. (2024). Using CNN for brain tumor diagnosis. In *Advances in medical diagnosis, treatment, and care (AMDTC) book series* (pp. 104–124). https://doi.org/10.4018/979-8-3693-3629-8.ch006

Lee, J., Mukhanov, L., Molahosseini, A. S., Minhas, U., Hua, Y., Del Rincon, J. M., Dichev, K., Hong, C., & Vandierendonck, H. (2023). Resource-Efficient Convolutional Networks: A survey on model-, Arithmetic-, and Implementation-Level techniques. *ACM Computing Surveys*, *55*(13s), 1–36. https://doi.org/10.1145/3587095

Maiboroda, R. (2024). Learning Languages in the 21st Century: the Digital and Ai Transformation. *International Journal of Social Science Humanity & Management Research*, *03*(11). https://doi.org/10.58806/ijsshmr.2024.v3i11n06

Ravindran, D., Pillai, R. H., Srinivasa, D., & Somashekaraiah, S. (2024). Service Quality Evaluation of Healthcare Sustainability: Patient Satisfaction, Service Delivery, and Clinical Outcomes. In *Service Quality Evaluation of Healthcare Sustainability: Patient Satisfaction, Service Delivery, and Clinical Outcomes* (pp. 57–76). https://doi.org/10.1007/978-3-031-48075-1_6

Rehman, S. U., Tu, S., Shah, Z., Ahmad, J., Waqas, M., Rehman, O. U., Kouba, A., & Abbasi, Q. H. (2021a). Deep learning Models for Intelligent Healthcare: implementation and challenges. In *Lecture notes in computer science* (pp. 214–225). https://doi.org/10.1007/978-3-030-78609-0_19

Savaglio, C., Valero, C. I., Belsa, A., Palau, C. E., & Fortino, G. (2023). Interoperability in cloud IoT platforms. In *Springer handbooks* (pp. 111–127). https://doi.org/10.1007/978-3-031-39650-2_6

Sedeeq, N. (2024). CNN-Based Segmentation and Detection of Brain Tumors MRI images: a review. *Indonesian Journal of Computer Science*, *13*(3). https://doi.org/10.33022/ijcs.v13i3.4029

T, S. K., V, A., E, S., & B, Y. (2024a). Automated brain tumor detection and Classification. *Journal of Soft Computing Paradigm*, *6*(4), 341–349. https://doi.org/10.36548/jscp.2024.4.001

Figure 3.6*: Showing A Comprehensive Guide to Classification Algorithms in Machine Learning (Arash hadad (2024) https://medium.com/pythoneers/a-comprehensive-guide-to-classification-algorithms-in-machine-learning-whats-the-point-2e87e21da4cf*

https://learnopencv.com/understanding-convolutional-neural-networks-cnn/

Figure 3.8*: Showing DevOps methodology and process Raycad (Nov 2028)*

https://medium.com/@raycad.seedotech/devops-methodology-and-process-dde388eb65bd

Figure 3.7: Showing Agile Methodology in System Development https://www.javatpoint.com/software-engineering-agile-model

Figure 2.2*: Showing Convolutional Neural Network Architecture Khuyen Le (March 2021) https://lekhuyen.medium.com/an-overview-of-vgg16-and-nin-models-96e4bf398484*

Figure 3.1*: Showing New Machine Learning Model Flags Abnormal Brain Scans in Real-time (February, 2022) https://www.itnonline.com/content/new-machine-learning-model-flags-abnormal-brain-scans-real-time*

APPENDIX - A

Research Study Activity Gantt Chat

Activity Label	Research Activities	Activity Mode of Execution	Dependent upon	Research Activities/Study Duration estimated per-week											
Integrated Research Methodology															
1. Qualitative Research Approach															
				1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
A	Proposal	Parallel	None												
B	Review of Related Literature	Dependent upon	A												
C	Schedule Development for Data gathering and participant selection	Parallel	B												
D	Pilot and final Data Collection phase	Dependent upon	A, B & C												
E	Analysing facts insights from the findings	Dependent upon	D												
F	write-up of the study report	Dependent upon	A, B, C, D & E												
Software Development Approach															
2. Agile Development Model															
I	Requirement gathering and Analysis	Parallel	None												
J	Design	Parallel	I												
K	Implementation (Coding/development)	Dependent upon	J												
L	Testing and Integration/Deployment	Dependent upon	K												
M	Maintenance	Dependent upon	I, J, K & L	Commence after the above stated activities are completed											

Source: *David Sapunka Fornah (January 2023)*

APPENDIX – B

Research Study Quad Chart

Problem statement and study Justification

The problem statement addresses the critical need for an automated brain tumor detection system using advanced machine learning techniques, specifically CNNs and U-Net algorithm, to enhance diagnostic accuracy, reduce human involvement, and improve healthcare service delivery by efficiently analyzing MRI scans with greater speed and precision compared to traditional manual methods.

Problem statement and study Justification

The problem statement addresses the critical need for an automated brain tumor detection system using advanced

Study Methodology

Qualitative Research Approach: The study gathers insights from stakeholders through qualitative methods to understand requirements, challenges, and expectations for the brain tumor detection system, which informed the development of the conceptual framework.

Software Development Approach: An agile methodology enhanced with DevOps practices was implemented, focusing on iterative development, continuous integration/deployment, and automated testing of the CNN-based tumor detection system. This approach enabled rapid prototyping and refinement of the U-Net algorithm implementation.

Aim and Objectives

To develop and deploy an automated brain tumor detection system that can enhance effective and efficient brain tumor diagnostics.

I. To analyze and evaluate the implementation of Convolutional Neural Networks (CNN) in healthcare service delivery systems, specifically for brain tumor detection.

II. To develop and validate a CNN modeling framework for automated brain tumor detection, including implementation procedures and optimization techniques.

III. To design and evaluate an effective user training program for healthcare professionals implementing CNN-based brain

Expected Outcome

The expected outcome is an automated brain tumor detection system that accurately analyzes MRI scans using CNN and U-Net algorithms, providing timely results with minimal human intervention. This system aims to streamline diagnostic processes in hospitals, reduce workload on healthcare professionals, and ultimately enhance patient care through faster and more accurate tumor detection.

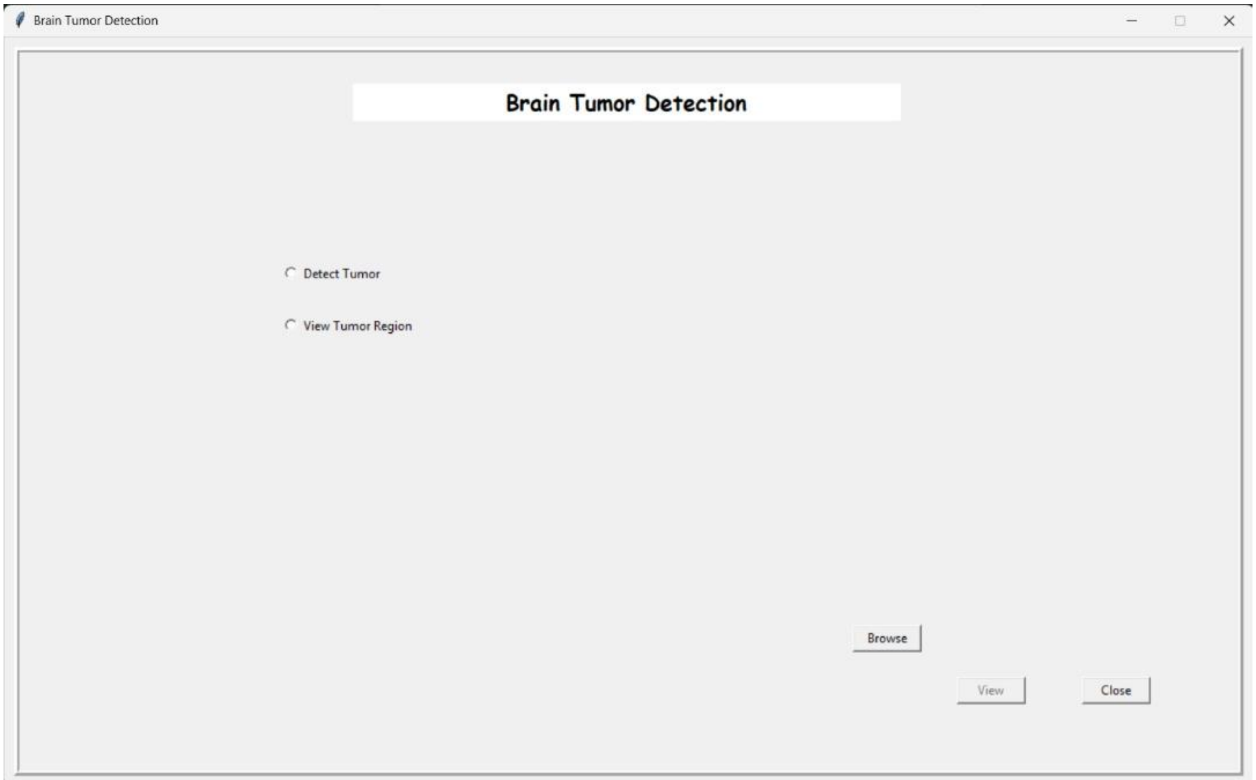
Expected Outcome

The expected outcome is an automated brain tumor

Source: Study material January (2025)

APPENDIX - C

Graphical User Interface




```
import tkinter
from PIL import Image
from tkinter import filedialog
import cv2 as cv
from frames import *
from displayTumor import *
from predictTumor import *


class Gui:
    MainWindow = 0
    listOfWinFrame = list()
    FirstFrame = object()
    val = 0
    fileName = 0
    DT = object()

    wHeight = 700
```

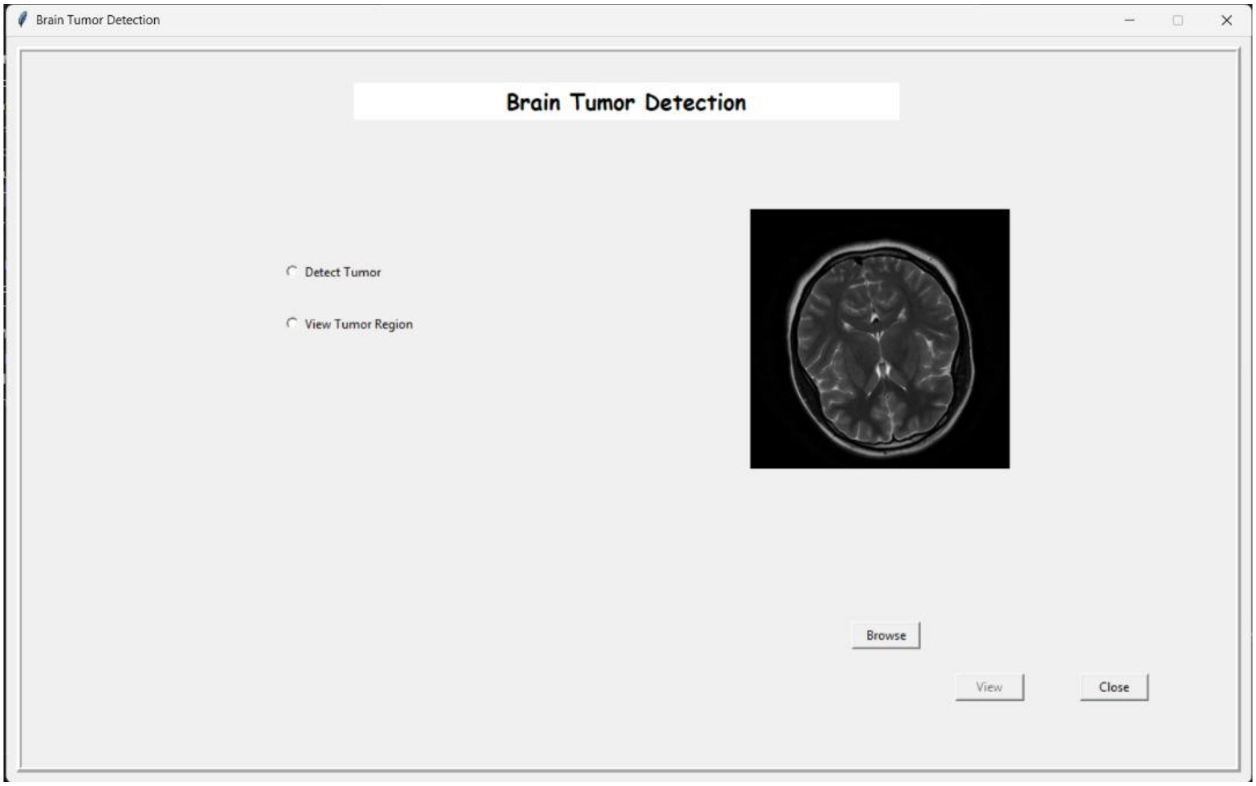
```
    wWidth = 1180

    def _init_(self):
        global MainWindow
        MainWindow = tkinter.Tk()
        MainWindow.geometry('1200x720')
        MainWindow.resizable(width=False, height=False)

        self.DT = DisplayTumor()

        self.fileName = tkinter.StringVar()

        self.FirstFrame = Frames(self, MainWindow, self.wWidth, self.wHeight, 0, 0)
        self.FirstFrame.btnView['state'] = 'disable'

        self.listOfWinFrame.append(self.FirstFrame)

        WindowLabel = tkinter.Label(self.FirstFrame.getFrames(), text="Brain Tumor Detection",
height=1, width=40)
        WindowLabel.place(x=320, y=30)
        WindowLabel.configure(background="White", font=("Comic Sans MS", 16, "bold"))

        self.val = tkinter.IntVar()
        RB1 = tkinter.Radiobutton(self.FirstFrame.getFrames(), text="Detect Tumor",
variable=self.val,
                                  value=1, command=self.check)
        RB1.place(x=250, y=200)
        RB2 = tkinter.Radiobutton(self.FirstFrame.getFrames(), text="View Tumor Region",
                                  variable=self.val, value=2, command=self.check)
        RB2.place(x=250, y=250)

        browseBtn = tkinter.Button(self.FirstFrame.getFrames(), text="Browse", width=8,
command=self.browseWindow)
        browseBtn.place(x=800, y=550)

        MainWindow.mainloop()

    def getListOfWinFrame(self):
        return self.listOfWinFrame

    def browseWindow(self):
        global mriImage
```

```
        FILEOPENOPTIONS = dict(defaultextension='.',
                               filetypes=[('jpg', '.jpg'), ('png', '.png'), ('jpeg', '.jpeg'), ('All Files', '.*')])
        self.fileName = filedialog.askopenfilename(**FILEOPENOPTIONS)
        image = Image.open(self.fileName)
        imageName = str(self.fileName)
        mriImage = cv.imread(imageName, 1)
        self.listOfWinFrame[0].readImage(image)
        self.listOfWinFrame[0].displayImage()
        self.DT.readImage(image)

    def check(self):
        global mriImage
        #print(mriImage)
        if (self.val.get() == 1):
            self.listOfWinFrame = 0
            self.listOfWinFrame = list()
            self.listOfWinFrame.append(self.FirstFrame)

            self.listOfWinFrame[0].setCallObject(self.DT)

            res = predictTumor(mriImage)

            if res > 0.5:
                resLabel = tkinter.Label(self.FirstFrame.getFrames(), text="Tumor Detected", height=1,
width=20)
                resLabel.configure(background="White", font=("Comic Sans MS", 16, "bold"), fg="red")
            else:
                resLabel = tkinter.Label(self.FirstFrame.getFrames(), text="No Tumor", height=1,
width=20)
                resLabel.configure(background="White", font=("Comic Sans MS", 16, "bold"),
fg="green")

            resLabel.place(x=700, y=450)

        elif (self.val.get() == 2):
            self.listOfWinFrame = 0
            self.listOfWinFrame = list()
            self.listOfWinFrame.append(self.FirstFrame)

            self.listOfWinFrame[0].setCallObject(self.DT)
            self.listOfWinFrame[0].setMethod(self.DT.removeNoise)
```

```
        secFrame = Frames(self, MainWindow, self.wWidth, self.wHeight, self.DT.displayTumor,
self.DT)

        self.listOfWinFrame.append(secFrame)


        for i in range(len(self.listOfWinFrame)):
            if (i != 0):
                self.listOfWinFrame[i].hide()
        self.listOfWinFrame[0].unhide()

        if (len(self.listOfWinFrame) > 1):
            self.listOfWinFrame[0].btnView['state'] = 'active'

    else:
        print("Not Working")

mainObj = Gui()
```

APPENDIX - D

Import Data Interface

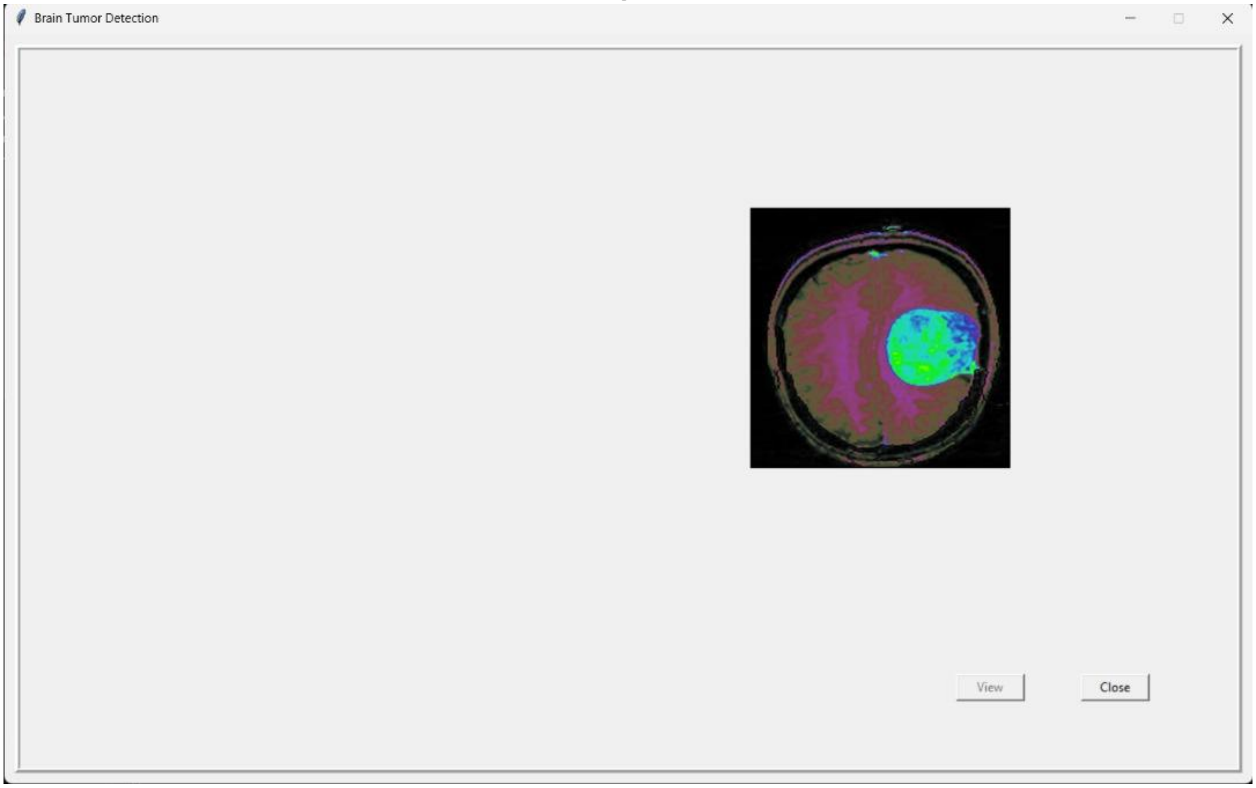



```
import numpy as np
import cv2 as cv


class DisplayTumor:
    curImg = 0
    Img = 0

    def readImage(self, img):
        self.Img = np.array(img)
        self.curImg = np.array(img)
        gray = cv.cvtColor(np.array(img), cv.COLOR_BGR2GRAY)
        self.ret, self.thresh = cv.threshold(gray, 0, 255, cv.THRESH_BINARY_INV + cv.THRESH_OTSU)

    def getImage(self):
        return self.curImg
```

```
# noise removal
def removeNoise(self):
    self.kernel = np.ones((3, 3), np.uint8)
    opening = cv.morphologyEx(self.thresh, cv.MORPH_OPEN, self.kernel, iterations=2)
    self.curImg = opening

def displayTumor(self):
    # sure background area
    sure_bg = cv.dilate(self.curImg, self.kernel, iterations=3)

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

    # Find unknown region
    sure_fg = np.uint8(sure_fg)
    unknown = cv.subtract(sure_bg, sure_fg)

    # Marker labelling
    ret, markers = cv.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 = cv.watershed(self.Img, markers)
    self.Img[markers == -1] = [255, 0, 0]

    tumorImage = cv.cvtColor(self.Img, cv.COLOR_HSV2BGR)
    self.curImg = tumorImage
```

APPENDIX - E

Tumor Region Interface




```
import tkinter
from PIL import ImageTk
from PIL import Image

class Frames:
    xAxis = 0
    yAxis = 0
    MainWindow = 0
    MainObj = 0
    winFrame = object()
    btnClose = object()
    btnView = object()
    image = object()
    method = object()
    callingObj = object()
    labelImg = 0

    def _init_(self, mainObj, MainWin, wWidth, wHeight, function, Object, xAxis=10, yAxis=10):
        self.xAxis = xAxis
```

```
        self.yAxis = yAxis
        self.MainWindow = MainWin
        self.MainObj = mainObj
        self.MainWindow.title("Brain Tumor Detection")
        if (self.callingObj != 0):
            self.callingObj = Object

        if (function != 0):
            self.method = function

        global winFrame
        self.winFrame = tkinter.Frame(self.MainWindow, width=wWidth, height=wHeight)
        self.winFrame['borderwidth'] = 5
        self.winFrame['relief'] = 'ridge'
        self.winFrame.place(x=xAxis, y=yAxis)

        self.btnClose = tkinter.Button(self.winFrame, text="Close", width=8,
                                      command=lambda: self.quitProgram(self.MainWindow))
        self.btnClose.place(x=1020, y=600)
        self.btnView = tkinter.Button(self.winFrame, text="View", width=8, command=lambda:
self.NextWindow(self.method))
        self.btnView.place(x=900, y=600)


    def setCallObject(self, obj):
        self.callingObj = obj


    def setMethod(self, function):
        self.method = function


    def quitProgram(self, window):
        global MainWindow
        self.MainWindow.destroy()


    def getFrames(self):
        global winFrame
        return self.winFrame
```

```
def unhide(self):
    self.winFrame.place(x=self.xAxis, y=self.yAxis)


def hide(self):
    self.winFrame.place_forget()


def NextWindow(self, methodToExecute):
    listWF = list(self.MainObj.listOfWinFrame)

    if (self.method == 0 or self.callingObj == 0):
        print("Calling Method or the Object from which Method is called is 0")
        return

    if (self.method != 1):
        methodToExecute()
    if (self.callingObj == self.MainObj.DT):
        img = self.MainObj.DT.getImage()
    else:
        print("Error: No specified object for getImage() function")

    jpgImg = Image.fromarray(img)
    current = 0

    for i in range(len(listWF)):
        listWF[i].hide()
        if (listWF[i] == self):
            current = i

    if (current == len(listWF) - 1):
        listWF[current].unhide()
        listWF[current].readImage(jpgImg)
        listWF[current].displayImage()
        self.btnView['state'] = 'disable'
    else:
        listWF[current + 1].unhide()
        listWF[current + 1].readImage(jpgImg)
        listWF[current + 1].displayImage()

    print("Step " + str(current) + " Extraction complete!")
```

```python
def removeComponent(self):
    self.btnClose.destroy()
    self.btnView.destroy()


def readImage(self, img):
    self.image = img


def displayImage(self):
    imgTk = self.image.resize((250, 250), Image.Resampling.LANCZOS)
    imgTk = ImageTk.PhotoImage(image=imgTk)
    self.image = imgTk
    self.labelImg = tkinter.Label(self.winFrame, image=self.image)
    self.labelImg.place(x=700, y=150)
```