FUSING TECHNOLOGIES FOR LITTLE LIVES MRI IMAGING IN INFANT BRAIN TUMOUR DETECTION

CO8811 – PROJECT REPORT

Submitted by

AMRIN REIHANA S	**211420118005**
ANUSHA FAMEL J	**211420118007**
HARSINI R.S	**211420118025**
MONISHA G	**211420118033**

in partial fulfillment for the award the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER AND COMMUNICATION ENGINEERING

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

MARCH 2024

BONAFIDE CERTIFICATE

Certified that this project report **"FUSING TECHNOLOGIES FOR LITTLE LIVES MRI IMAGING IN INFANT BRAIN TUMOUR DETECTION"** is the bonafide work of **AMRIN REIHANA S (211420118005), ANUSHA FAMEL J (211420118007), HARSINI R.S (211420118025) AND MONISHA G (211420118033)** who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Dr.B.ANNI PRINCY M.E., Ph.D.,	J.BRINDHA M.E., (Ph.D.).,
HEAD OF THE DEPARTMENT	**SUPERVISOR**
PROFESSOR, COMPUTER AND COMMUNICATION ENGINEERING, PANIMALAR ENGINEERING COLLEGE, NAZARATHPETTAI, POONAMALLEE, CHENNAI- 600123.	ASSISTANT PROFESSOR, ELECTRONIC AND COMMUNICATION ENGINEERING, PANIMALAR ENGINEERING COLLEGE, NAZARATHPETTAI, POONAMALLEE, CHENNAI- 600123.

Certified that the above candidate(s) was/ were examined in the End Semester

Project Viva-Voce Examination held on..........................

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

We express our deep gratitude to our respected Secretary and Correspondent **Dr.P.CHINNADURAI, M.A., Ph.D.** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We would like to extend our heartfelt and sincere thanks to our Directors Tmt. **C.VIJAYARAJESWARI, Dr. C. SAKTHIKUMAR, M.E., Ph.D**., and **Dr.SARANYASREE SAKTHIKUMAR B.E., M.B.A., Ph.D**., for providing us with the necessary facilities for completion of this project.

We also express our gratitude to our Principal **Dr.K.MANI, M.E., Ph.D.,** for his timely concern and encouragement provided to us throughout the course.

We thank our HOD of Computer and Communication Engineering Department, **Dr. B. ANNI PRINCY, M.E., Ph.D**., Professor, for the support extended throughout the project.

We would like to thank our supervisor, **Mrs. J. BRINDHA, M.E., (Ph.D.)**., Assistant Professor, and all the faculty members of the Department of Electronic and Communication Engineering for their advice and suggestions for the successful completion of the project.

AMRIN REIHANA S

ANUSHA FAMEL J

HARSINI R.S

MONISHA G

ABSTRACT

The detection of infant brain abnormalities is a critical step in paediatric care, as early detection and diagnosis can lead to better outcomes for both the infant and the caregivers. Currently, infant brain abnormalities are primarily detected using various imaging techniques, including ultrasound, MRI, and CT scans. However, these methods can be limited in their ability to provide detailed information about the developing brain, particularly in real-time scenarios. In this work, we propose a deep learning-based approach for the detection and classification of infant brain abnormalities using medical imaging data. The proposed approach utilizes a Convolutional Neural Network (CNN) trained on a large dataset of medical images annotated with infant brain abnormalities. The CNN is trained to detect and classify abnormalities in real-time, providing a faster and more accurate method of identifying potential issues during infancy. Experiments conducted on a diverse dataset demonstrate the effectiveness of the proposed approach, with high accuracy and sensitivity in detecting infant brain abnormalities. The results suggest that the proposed deep learning-based approach has the potential to significantly improve paediatric care. With its ability to analyze vast amounts of medical imaging data quickly and accurately, this technology has the potential to streamline diagnostic workflows and reduce the burden on healthcare professionals, ultimately improving the overall efficiency and quality of paediatric care. In summary, the proposed deep learning-based approach represents a promising advancement in the early detection and classification of infant brain abnormalities with an accuracy of 95%, with far-reaching implications for paediatric medicine and the well-being of infants worldwide.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVATION

S.NO	ABBREVATION	Expansion
1	DWT	Discrete wavelet transform
2	CNN	Convolutional neural network
3	MRI	Magnetic resonance imaging
4	DWI	Diffusion- weighted imaging
5	SVM	Support vector machine
6	KNN	K nearest neighbor
7	MRS	Magnetic resonance Spectroscopy
8	PWI	Perfusion weighted imaging
9	AI	Artificial intelligence
10	ANN	Artificial neural network
11	GUI	Graphics user interface
12	SGD	Stochastic gradient descent algorithm
13	RAM	Random access memory

CHAPTER 1

INTRODUCTION

This work delves into the complex field of pediatric neuroimaging, with a particular emphasis on the use of magnetic resonance imaging (MRI) to identify brain tumors in newborns at an early stage. It explores the difficulties associated with scanning infants, the developments in MRI technology, and the therapeutic ramifications of combining these technologies with pediatric oncology. This research tries to clarify the revolutionary potential of merging technologies for these fragile little lives through a thorough analysis of recent literature, technical advancements, and therapeutic practices.

Few breakthroughs in the field of medical advancements are as complex and nuanced as pediatric neuroimaging. This field is where the fragility of small lives and technical progress meet. [1,9,13,25,32] The early diagnosis of brain tumors in babies, where every second matters in determining treatment results and prognosis, is one of the most urgent concerns facing doctors or other medical practices.

As a fundamental tool for pediatric neuroimaging, magnetic resonance imaging (MRI) provides unmatched insights into the complex architecture of the growing brain. But diagnosing brain tumors in newborns poses special difficulties that call for a sophisticated strategy combining state-of-the-art technology and medical knowledge.

In order to better serve these small lives, this study will investigate the field of combining technologies. Specifically, it will concentrate on the use of MRI imaging to detect brain tumors in infants early on. We hope to shed light on the revolutionary potential of this multidisciplinary approach through a thorough examination of the current state of affairs, technical developments, and clinical consequences.

At the core of this conversation is a fundamental realization: the combination of technologies not only improves our capacity to perceive the invisible, but it also represents a ray of hope for countless families facing the terrifying prognosis of brain tumors in infants. As we set out on this path that unites science and compassion, let us imagine a moment in the future when every infant has access to early identification, prompt care, and a route to recovery.

Because of their small size, quickly developing brains, and limited ability to sit still during imaging operations, infants present special problems in the field of neuroimaging. Infants may not benefit from standard MRI procedures meant for adults, necessitating the use of equipment and specific protocols.

Furthermore, a sophisticated comprehension of both abnormal and typical developmental variances is necessary for the interpretation of MRI data in babies. This complex terrain requires radiologists and pediatric neurologists to identify with accuracy and skill between potentially fatal tumors and benign developmental abnormalities.

Recent years have seen a great deal of development in MRI technology [5,10,17,18,22,24], which has improved its use in pediatric neuroimaging. Additional layers of information are provided by methods including diffusion-weighted imaging (DWI), magnetic resonance spectroscopy (MRS), and perfusion-weighted imaging (PWI), which help distinguish between different types of tumors, determine how aggressive they are, and direct treatment decisions. By measuring the random movement of water molecules inside tissues, diffusion-weighted imaging (DWI) can reveal information on the cellular density and microstructure of the tissue. Based on variations in cellularity and membrane integrity, DWI in brain tumor imaging can assist in differentiating between tumor and normal brain tissue [4].

Tissue biochemistry can be studied by the non-invasive monitoring of metabolite concentrations in tissues using magnetic resonance spectroscopy (MRS). MRS can be used in brain tumor imaging to track variations in metabolite levels over time in order to assess treatment response and characterize different types of tumors based on their metabolic profiles.

By measuring the amount of contrast agent that passes through the vasculature, perfusion-weighted imaging, or PWI, assesses the perfusion of tissue. PWI in brain tumor imaging can help determine tumor aggressiveness and treatment response by offering information on tumor vascularity, blood-brain barrier permeability, and tissue viability.

Beyond conventional MRI methods, the use of artificial intelligence (AI) has the potential to improve pediatric neuroimaging's diagnostic potential. Large volumes of imaging data may be analyzed by machine learning algorithms, whichcan then be used to spot minor anomalies and make more accurate patient outcome predictions. Integrating MRI imaging with other diagnostic modalities and clinical experience is crucial in the field of pediatric cancer. To guarantee a thorough approach to patient care, multidisciplinary teams including radiologists, pediatric neurologists, neurosurgeons, oncologists, and other professionals work together.

Technology for diagnosing brain tumors in infants is a result of the combination of clinical knowledge, scientific advancement, and humane treatment. By combining cutting-edge MRI methods with artificial intelligence and interdisciplinary teamwork, medical professionals have the potential to significantly improve the early identification and treatment of brain tumors in young patients, providing a glimmer of hope to innumerable families across the globe [6].

Let us not waver in our dedication to the welfare of these tiny beings as we negotiate the challenges of pediatric neuroimaging. We can create a future where every child with a brain tumor receives prompt intervention, knowledgeable direction, and unwavering support on their path to recovery by embracing the transformative potential of technology and encouraging collaboration across disciplines.

1.1 SCOPE OF THE PROJECT

The project aims to develop advanced methods for accurately detecting brain tumors in infants using MRI images, focusing on deep learning- based fusion techniques. It involves comprehensive literature review, data collection, and preprocessing to ensure quality and consistency. Deep learning models tailored for infant brain tumor detection will be developed, exploring fusion techniques to combine features from multiple sources for enhanced accuracy. Evaluation metrics will be defined, with validation and testing on separate datasets to ensure generalization. Ethical considerations regarding patient data privacy and regulatory compliance will be addressed. The project willculminate in a thorough analysis comparing the performance of fused deep learning models with baseline methods, with emphasis on clinical relevance. Findings will be documented in a comprehensive report or academic paper, highlighting implications for medical practitioners.

CHAPTER 2

LITERATURE SURVEY

2.1 OVERVIEW

Title 1: Brain tumor detection using CNN, Alex Net & Google Net ensemble learning approaches

Author: Chetan Swarup, Kamred Udham Singh, Ankit Kumar 3, Saroj Kumar Pandey, Neerajvarshney and Teekam Singh

Year: 2023

Description:

Our study proposes a deep CNN model for brain tumor detection in MRI images, employing Google Net and Alex Net architectures. Preprocessing generates convolutional features capturing tumor patterns. Evaluation metrics include accuracy, sensitivity, specificity, and AUC. Google Net's efficiency, with fewer parameters and computational resources, makes it preferable for real-world applications. Our CNN model, especially when based on Google Net, demonstrates high accuracy, sensitivity, specificity, and computational efficiency. It presents as a promising support tool for radiologists, enhancing brain tumor detection and potentially improving patient outcomes.

Title 2: Convolution neural network-based automatic localization of landmarks on lateral X-ray images

Author: R. A. Ramadan, A. Y. Khedr, K. Yadav, E. J. Alreshidi, M. H. Sharif, A. T. Azar, and H. Kamberaj

Year: 2022

Description:

Cephalometric analysis is essential for patients having dentofacial and craniofacial deformities. The manual localization of the cephalometric landmarks is also important and critical for the observer which is required to be performed by the orthodontics only. The proposed method is a deep learning approach where centroid-based registration was performed on the same size images then ResNet50 was applied on the different patches which were made based on the geometrical position of the landmarks. A total of ten patches were made for the 19 landmarks.

Title 3: A comprehensive survey on brain tumor diagnosis using deep learning and emerging hybrid techniques with multi-modal MR image

Author: S. Ali, J. Li, Y. Pei, R. Khurram, K. U. Rehman, and T. Mahmood

Year: 2022

Description:

Brain tumor is considered the deadly disease of the century. At present, neuroscience and artificial intelligence conspire in the timely delineation, detection, and classification of brain tumors. Therefore, there is an essential requirement to build computer-aided diagnosis systems to diagnose brain tumors timely. We have summarized the work of researchers published in the last decade termed as the 10s and the present decade termed as the 20s. The decades in review reveal the bore witness to the critical revolutionary paradigm shift in artificial intelligence viz. conventional/machine learning methods, emerged deep learning, and emerging hybrid techniques.

Title 4: Brain Tumor Detection Using U-Net and 3D CNN Architecture

Author: Anshul Ghanshala, Aakarshan Chauhan, Manoj Diwakar; Sachin Sharma.

Year: 2022

Description:

The paper discusses the challenge of manually segmenting brain tumors from MRI data and presents various techniques for automating this process. It introduces a novel architecture that combines a 3D Convolutional Neural Network (CNN) and a U-Net for brain tumor detection. The 3D CNN extracts feature from MRI scans, while the U-Net refines the segmentation to highlight tumor regions. Evaluation against existing methods shows superior performance, indicating the effectiveness of the proposed architecture. However, further research is needed to validate its applicability across different datasets and clinical settings. Overall, the paper provides a comprehensive overview of brain tumor detection techniques in MRI images.

Title 5: Convolution neural network-based automatic localization of landmarks on lateral X-ray images

Author: R. A. Ramadan, A. Y. Khedr, K. Yadav, E. J. Alreshidi, M. H. Sharif, A. T. Azar, and H. Kamberaj

Year: 2022

Description:

Cephalometric analysis utilizes deep learning to automatically localize landmarks on images of patients with dentofacial and craniofacial deformities, reducing manual effort for orthodontists. The process begins with centroid-based registration, aligning images based on reference points. Images are then divided into patches around cephalometric landmarks. ResNet50, a CNN known for image recognition, analyzes these patches individually to learn features and localize landmarks accurately. Evaluation likely includes metrics like accuracy, precision, and recall using annotated datasets. Iterative refinementmay enhance accuracy and robustness across different images and populations. This method improves efficiency and accuracy in orthodontic diagnosis and treatment planning by automating landmark localization.

Title 6: Image Processing based Brain Tumor Detection

Author: Shivam Varshney; Shubham Kumar Prajapati; Sahil Rajput; Mandeep Kaur; Nitin Rakesh; Mayank Kumar Goyal.

Year: 2021

Description:

The importance of MRI in brain tumor detection, emphasizing the need for image processing techniques to enhance accuracy. It outlines key steps including pre-processing, segmentation, feature extraction, and classification using Support Vector Machine (SVM). SVM classification with extracted features is shown to offer superior accuracy in identifying tumor regions compared to other methods. However, further research is needed to improve sensitivity and specificity for real-world clinical applications. Overall, the study underscores the value of image processing as a valuable tool in brain tumor diagnosis, highlighting the need for a multidisciplinary approach in treatment.

Title 7: Brain Tumor Classification Using Deep Learning Technique -- A Comparison between Cropped, Uncropped, and Segmented Lesion Images with Different Sizes

Author: A. M. Alqudah, H. Alquraan, I. A. Qasmieh, A. Alqudah, and W. Al-Sharu

Year: 2020

Description:

As a proven powerful machine learning tool, deep learning was widely used in several applications for solving various complex problems that require extremely high accuracy and sensitivity, particularly in the medical field. In general, brain tumor is one of the most common and aggressive malignant tumor diseases which is leading to a very short expected life if it is diagnosed at higher grade. Based on that, brain tumor grading is a very critical step after detecting the tumor in order to achieve an effective treating plan.

Title 8: Brain tumor detection by using stacked autoencoders in deep learning

Author: J. Amin, M. Sharif, N. Gul, M. Raza, M. A. Anjum, M. W. Nisar, and S. A. C. Bukhari

Year: 2020

Description:

In this manuscript, a deep learning model is deployed to predict input slices as a tumor /non-tumor. This manuscript employs a high pass filter image to prominent the inhomogeneities field effect of the MR slices and fused with the input slices. Moreover, the median filter is applied to the fused slices. The resultant slices quality is improved with smoothen and highlighted edges of the input slices. After that, based on these slices' intensity, a 4-connected seed growing algorithm is applied, where optimal threshold clusters the similar pixels from the input slices. The segmented slices are then supplied to the fine-tuned two layers proposed stacked sparse autoencoder model.

Title 9: Brain Tumor Segmentation Using Deep Learning and Fuzzy K-Means Clustering for Magnetic Resonance

Author: R. Pitchai, P. Supraja, A. Helen Victoria and M. Madhavi

Year:2020

Description:

The primary objective of this paper is to develop a methodology for brain tumor segmentation. Nowadays, brain tumor recognition and fragmentation are one among the pivotal procedure in surgical and medication planning arrangements. It is difficult to segment the tumor area from MRI images due to inaccessibility of edge and appropriately visible boundaries. In this paper, a combination of Artificial Neural Network and Fuzzy K-means algorithm has been presented to segment the tumor locale. It contains four phases, Noise evacuation Attribute extraction and selection Classification and Segmentation.

Title 10: Brain tumor detection based on segmentation using MATLAB.

Author: Animesh Hazra, Ankit Dey, Sujit Kumar Gupta, Md. Abid AnsariYear:2017

Description:

The paper introduces a methodology for brain tumor detection in MRI images, comprising pre-processing, edge detection, and segmentation stages. Pre-processing involves noise removal and grayscale conversion to enhance image quality. Edge detection algorithms like Sobel, Prewitt, and Canny identify tumor boundaries. Segmentation isolates tumor-affected regions, crucial for accurate localization. The k-means clustering algorithm further processes segmented images to group similar pixels, aiding in tumor boundary delineation. MATLAB is utilized for algorithm development, leveraging its extensive image processing capabilities for medical image analysis.

2.2 EXISTING SYSTEM

There are several existing solutions to detecting and classifying fetal brain abnormalities. One of the widely used approaches is the use of ULTRASOUND scans. However, ULTRASOUND is an expensive and time-consuming process and can be uncomfortable for the patient. Another solution is ultrasound imaging, which is less expensive and less invasive but has limited image resolution.

Some studies have employed K-means clustering algorithms to segment MRI images of infant brains into distinct regions, potentially identifying areas of abnormal tissue that could indicate the presence of a tumor. This method involves iteratively partitioning the image data into K clusters based on similarity measures, aiming to separate tumor regions from healthy brain tissue. Additionally, K-means clustering may struggle with overlapping intensity distributions in MRI images, leading to inaccuracies in tumor delineation.

Other existing solutions include manual analysis allows for detailed scrutiny of image features and can leverage the expertise of experienced practitioners, it is time-consuming, subjective, and prone to interobserver variability.

However, all these existing solutions have their limitations, such as the need for large amounts of annotated data, high computational cost, and difficulty in capturing complex relationships between features. These limitations hinder the performance of these methods and restrict their application in real-world scenarios. In addition to the existing solutions, other methods have been proposed for detecting and classifying fetal brain abnormalities.

These include machine learning algorithms such as random forests, decision trees, and support vector machines (SVM), as well as deep learning methods such as convolutional neural networks (CNN) and recurrent neural networks (RNN).

When making predictions, it calculates the distance between the input data point and all the training examples, using a chosen distance metric such as Euclidean distance. Next, the algorithm identifies the K nearest neighbour to the input data point based on their distances. In the case of classification, the algorithm assigns the most common class label among the K neighbour as the predicted label for the input data point. For regression, it calculates the average or weighted average of the target values of the K neighbour to predict the value for the input data point.

However, deep learning methods have thresholding techniques involve setting intensity thresholds to segment MRI images, separating regions of interest, such as tumors, from background tissue. These methods can range from simple global thresholding to more sophisticated adaptive thresholding approaches. While thresholding methods offer a straightforward way to automate segmentation, they can be sensitive to variations in image intensity, noise, and imaging artifacts.

Overall, there is still much research to be done in this field, and ongoing efforts to improve the accuracy and robustness of methods for detecting and classifying fetal brain abnormalities are crucial for improving patient outcomes.

2.2.1 PROBLEM IDENTIFICATION

- Manual analysis of MRI images for infant brain tumor detection takes a significant amount of time, which can delay diagnosis and treatmentinitiation.
- Manual analysis may not be feasible for handling large volumes of imagingdata or for real-time diagnosis, limiting its applicability in clinical practice.
- Reliance on predefined intensity thresholds may not capture image variability.
- Determining appropriate threshold values can be challenging, especially with heterogeneous tumors. Oversimplification of tumor detection may lead to inaccurate segmentation results.

- Variability in tumor shapes and sizes may not align well with clustering assumptions and also can lead to inconsistent results

2.2.2 LIMITATIONS

- Existing solutions for detecting and classifying fetal brain abnormalities require a large amount of data to train the algorithms. However, limited availability of data often leads to suboptimal performance.
- Algorithms may struggle to distinguish between different types of abnormalities, leading to issues such as over diagnosis (detecting false positives) or under diagnosis (missing important anomalies). This limitation can impact the accuracy and reliability of diagnoses.
- The results provided by current algorithms may lack consistency, leading to variations in the diagnosis of fetal brain abnormalities. Inconsistent results can undermine confidence in the accuracy of the diagnoses.
- Many existing solutions have high computational requirements, including significant resources and time. This complexity makes them impractical for routine use in clinical settings, where efficiency and accessibility are crucial.
- MRI imaging equipment and facilities are costly to acquire and maintain, which can limit accessibility, especially in resource-constrained healthcare settings.
- Limited availability of labeled data for training, especially for rare tumor types or specific patient populations, can hinder the development and evaluation of robust detection models.

CHAPTER 3

SYSTEM DESIGN

In the domain of medical diagnostics, the accurate and timely detection of brain tumors, particularly in infants, stands as a critical challenge with profound implications for patient care. Leveraging the power of deep learning and Convolutional Neural Networks (CNNs), this study embarks on a journey to revolutionize infant brain tumor detection. By harnessing the capabilities of CNNs in analyzing complex medical imaging data, this research endeavors to automate the classification of brain diseases with unprecedented accuracy and efficiency. This paper presents a comprehensive framework encompassing key stages of dataset acquisition, image preprocessing, feature extraction, and model training and evaluation. Through meticulous experimentation and validation, the proposed system aims to deliver reliable and actionable insights, empowering healthcare professionals with advanced diagnostic tools for early intervention and improved patient outcomes. By elucidating the intricacies of CNN-based brain tumor detection, this study aspires to contribute to the frontier of medical imaging analysis, paving the way for transformative advancements in clinical practice and healthcare delivery. Additionally, the study investigates innovative approaches for interpreting model predictions, including attention mechanisms and saliency mapping, to provide clinicians with insightful explanations and confidence intervals for diagnostic decisions. Through these endeavors, the research aims to push the boundaries of infant brain tumor detection, ushering in a new era of precision medicine and personalized healthcare interventions.

3.1 PROPOSED SYSTEM ARCHITECTURE DESIGN

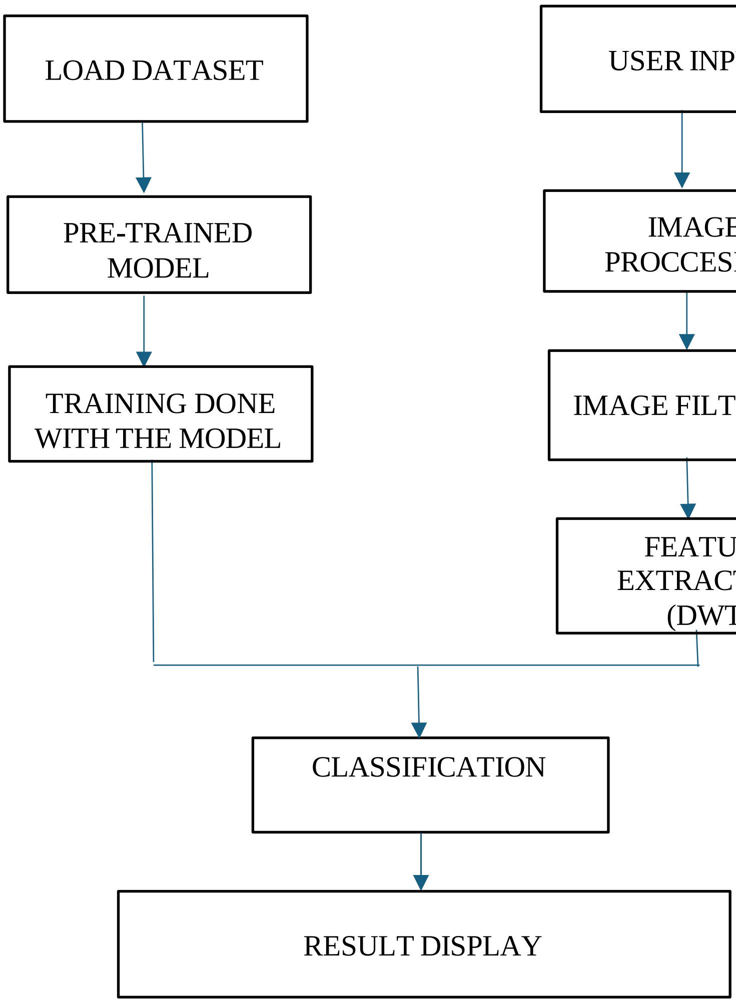

Figure 3.1 System Architecture

In addition to the foundation architectural components, as shown in figure 3.1 the research aims to push the boundaries of infant brain tumor detection, ushering in a new era of precision medicine.

3.1.1 DATASET

Upon clicking the "Load Dataset" button, the system accesses a specified directory containing the training dataset sourced from the Open Access Series of Imaging Studies OASIS repository, primarily comprising MRI scans of infant brain tumors. Subsequently, a global variable named "train" is instantiated as an image datastore object, enabling efficient management of the dataset. A notification message promptly notifies the user of the successful dataset loading, facilitating seamless subsequent processing and analysis.

3.1.2 PRE-TRAINED MODEL

In this project, the pre-trained models for detecting brain tumors in infants. These models come already equipped with knowledge from being trained on lots of different images. By using that, we save time and computer power because they’re already good at spotting patterns in new images, like infant brain scans. It adjusts these models to work specifically for the project, making them even better at finding tumors in infant brains. So, by using pre-trained models, it can diagnose and treat brain tumors in infants more quickly and accurately.

Input Layer

The input layer acts as the starting point for data entering the neural network. Its main job is to handle the incoming data, setting its dimensions as 120 pixels wide, 120 pixels tall, with 3 channels. It also ensures compatibility with pre-trained models and begins the process of extracting important features from the data. By specifying how the data should look and guiding its flow through the network, the input layer is crucial for accurate classification.

Fully Connected Layer

In the infant brain tumor detection setup, the fully connected layer with 20 neurons is added to the neural network design. It helps by making the data simpler to handle while keeping important details, making it easier for the network to understand tricky patterns, and making sure it can classify images accurately. With its 20 neurons, this layer strikes a balance between model complexity and generalization capability, contributing significantly to accurate classification of infant brain tumor images.

Softmax Layer

The softmax layer is an essential part of the neural network configuration in this project. This layer is activated after a pre-trained GoogLeNet model is loaded. Its task is to convert the network's raw predictions into probabilities for every conceivable class for the given dataset. Upon feeding an image into the network, the softmax layer assists in determining the likelihood that it falls into certain categories. The model determines which class the image most likely belongs to by analyzing these probabilities. To put it another way, the softmax layer facilitates the easier and more accurate classification of images by converting the network's complicated predictions into probabilities that are easy to comprehend.

Classification Layer

The classification layer is crucial in our project because it helps our CNN accurately sort MRI images of infant brain tumors. It has two classes such as normal and abnormal. It ensures that the network's output is suitable for this task, calculates loss during training to improve the model's learning, and evaluates the model's performance afterward. With its ability to classify images into multiple categories, it helps us identify different types of tumors and their characteristics. Overall, the classification layer ensures our CNN works effectively in detecting and diagnosing infant brain tumors early.

3.1.3 TRAINING

Training is essential in this project to enable the neural network, specifically the GoogleNet architecture, to learn and extract meaningful features from the provided dataset of infant brain tumor MRI images. The ratio of training to testing is 80:20. By exposing the network to a large collection of labeled images and adjusting its internal parameters using optimization algorithms like Stochastic Gradient Descent with Momentum (SGDM), the network can effectively differentiate between normal and abnormal brain tumor images. Thistraining process allows the network to generalize patterns and relationships from the training data to new, unseen data, ultimately contributing to accurate classification and early detection of infant brain tumors, thereby improving medical diagnosis and patient care.

Epoch

In the training setup, epochs play a crucial role in refining the performance of the neural network. Each epoch signifies one complete iteration through the training dataset. By training for 10 epochs, it provides the network with multiple opportunities to learn from the data and adjust its parameters to minimize training loss. This iterative process enables the network to capture complex patterns and improve its ability to generalize to new, unseen data. Overall, the use of epochs enhances the accuracy and effectiveness of this model in classifying infant brain tumor MRI images.

Batch size

The chosen mini-batch size of 20 facilitates efficient and effective model training. It enables frequent weight updates, parallel processing, and memory management, contributing to faster convergence and better generalization performance.

Learning rate

In this training process, the learning rate serves a crucial role in controlling how quickly or slowly the neural network adjusts its weights to minimize the loss function. It sets an initial learning rate of 0.001 and use a piecewise schedule with a drop factor of 0.1 every 8 epochs to balance convergence speed and stability. This ensures effective learning from the training data while preventing convergence issues.

Drop Factor

In this training process, the drop factor in the learning rate schedule controls the magnitude of learning rate adjustment when it drops at predefined intervals, set at 0.1 in drop factor case. This adaptive adjustment helps fine-tune the model’s convergence towards an optimal solution by preventing oscillations or divergence in the optimization process, ensuring stable and efficient training of the neural network.

L2 regularization

L2 regularization is used in the training method to penalize big weights in the network, hence preventing overfitting. By using this regularization strategy, generalization to previously unseen data is improved and model simplicity is maintained. It achieves a balance between fitting the training data well and guaranteeing good performance on new data by integrating L2 regularization with a given strength, like 0.004.

Iteration

In the training process, choosing 100 iterations for the training procedure enables the neural network to learn a great deal from the dataset over several runs. With more iterations, the model is better able to identify complex patterns in the data, which could result in enhanced accuracy and performance.

Stochastic Gradient Descent with Momentum

Stochastic Gradient Descent with Momentum (SGDM) optimization is selected for its efficacy in training deep convolutional neural networks in the project. SGDM enhances the standard SGD algorithm by incorporating a momentum term, which accelerates convergence and reduces oscillations during training. This optimization method is particularly suitable for the task of classifying MRI images of infant brain tumors as it aids the model in navigating complex architectures more efficiently. By fine-tuning parameters such as the initial learning rate and regularization terms, SGDM optimizes the training process, leading to improved stability and convergence speed. Ultimately, SGDM enables effective training of the CNN to accurately classify brain tumor images, contributing to the project's success.

3.1.4 USER INPUT

This function is associated with the "Select Image" button. It allows the user to choose an image file, i.e. BMP, PNG, or JPG format for classification. Then the selected image is load and displayed in the GUI. In deep learning, "user input" refers to data provided by users for training, configuring, and interacting with models. It includes labelled training data, hyperparameter settings, evaluation datasets, real-time interactive input, additional data for fine-tuning, and user feedback for model improvement. Managing user input is crucial for developing effective and user-friendly deep learning applications.

3.1.5 IMAGE PROCESSING

In this project, image processing serves the primary purpose of preprocessing MRI images of infant brains before feeding them into the deep learning model for tumor detection. The main goals include enhancing the quality of the images, extracting relevant features, and reducing noise to improve the model's accuracy in identifying brain tumors. Specifically, techniques like converting images to grayscale and performing edge detection help highlight important features and edges that are indicative of potential tumors. By incorporating image processing, the aim is to enhance the performance of the deep learning model and enable more accurate classification of brain tumor images.

Grayscale

In the project, grayscale is utilized by transforming MRI images of infant brains into simpler single-channel representations based on intensity, typically ranging from 0 (black) to 255 (white). This conversion helps streamline the computational analysis and allows a more effective focus on the structural aspects of the images. By examining these intensity variations within the grayscale range, the model can accurately detect abnormalities, such as tumors, leading to better outcomes in medical imaging diagnostics.

Edge Detection

The Sobel algorithm was chosen for edge detection in MRI images of infant brains due to its effectiveness in highlighting important features, particularly edges, which are crucial for tumor detection. The Sobel algorithm excels at detecting both vertical and horizontal edges, making it well-suited for capturing the intricate structures present in medical images. Its simplicity and efficiency make it an attractive choice for real-time processing, which is essential for medical diagnostic applications. By accurately identifying edges, the Sobel algorithm enhances the model's ability to detect abnormalities, such as tumors, with greater precision. Additionally, Sobel edge detection helps reduce noise and enhance image quality, leading to more reliable diagnostic results. Overall, the Sobel algorithm plays a vital role in improving the performance and accuracy of tumordetection in infant brain MRI scans, making it a preferred choice in this project.

3.1.6 IMAGE FILTERING

Image filtering is integral to the project for enhancing MRI image quality and improving tumor detection in infant brains. By removing noise and highlighting important features, such as tumor regions, filtering techniques like median filtering enhance image clarity and standardize quality across datasets. This ensures consistent analysis and reliable diagnostic outcomes, ultimately improving the effectiveness of the tumor detection model.

Median Filtering

Median filtering is utilized in this project to effectively remove noise from MRI images of infant brains, thereby enhancing the clarity of important features, such as tumor regions. The main purpose of median filtering is to replace each pixel's intensity value with the median value of neighboring pixels. This process effectively reduces the impact of outliers or random fluctuations in pixel values, resulting in smoother images with preserved edge details. In this project, median filtering is applied to the loaded MRI images as a preprocessing step before tumor detection. By smoothing out noise while preserving critical image features, median filtering improves the accuracy and reliability of tumor detection algorithms, ultimately leading to more precise diagnostic outcomes.

3.1.7 FEATURE EXTRACTION

Feature extraction is a critical component of the project, essential for identifying key patterns and attributes from MRI images of infant brains, particularly for tumor detection. By transforming raw image data into a concise representation, methods like convolutional neural networks (CNNs) and Discrete Wavelet Transform (DWT) automatically learn and extract distinctive features

indicative of tumor presence or characteristics. This process enables accurate classification and localization of tumors, aiding healthcare professionals in diagnosing and treating infants with brain abnormalities. Additionally, feature extraction simplifies the development of robust tumor detection algorithms by reducing data complexity and focusing on significant image attributes, ultimately enhancing the accuracy and efficiency of tumor detection in infant brain MRI scans.

Discrete Wavelet Transform

The Discrete Wavelet Transform (DWT) is a technique utilized in this project to analyze MRI images of infant brains. It breaks down the images into different frequency components, helping to identify important details like textures and structures that could indicate the presence of tumors. This method also aids in reducing the complexity of the data while keeping essential information intact and helps clean up any noise in the images. With DWT, tumor scan be detected more accurately across various scales within infant brains, making the project more effective in diagnosing brain abnormalities in infants using MRI scans.

3.1.8 CLASSIFICATION

The proposed work classification is to accurately categorize MRI images of infant brains as either normal or abnormal, particularly in terms of the presence or absence of tumors. By training classification models using deep learning techniques, such as convolutional neural networks (CNNs), the project aims to automate the process of identifying brain abnormalities in infants. This automated classification assists healthcare professionals in diagnosing and treating conditions early, improving patient outcomes and potentially saving lives. Additionally, classification allows for the efficient analysis of large volumes of medical imaging data, facilitating faster and more reliable diagnoses.

3.1.9 PERFORMANCE METRICS

Performance metrics for brain tumor detection refer to the measures used to evaluate the effectiveness and accuracy of algorithms or systems designed to identify brain tumors from medical imaging data such as MRI (Magnetic Resonance Imaging) scans. These metrics are crucial for assessing the reliability and efficiency of the detection process. To evaluate the deep learning models and analyze their performances, some metrics such as the accuracy, recall, precision and F1 score.

Accuracy

Accuracy represents the percentage of correctly identified cases of brain tumors among all infants examined, offering a straightforward evaluation of the model's effectiveness. Accuracy measures the number of correct predictions divided by the total number of samples. Applying the equation below, accuracy can be calculated, providing a measure of the model's performance.

Accuracy = ((Number of Correct Predictions)/ (Total Number of Predictions)) × 100%

To find Number of Correct predictions:

Number of Correct Prediction = Accuracy × Total Number of Predictions

Recall

Recall, specifically in the realm of infant brain tumor detection, refers to the model's capability to correctly identify all actual cases of brain tumors among infants from the entire set of brain tumor cases present. It focuses on avoiding missing any positive instances as false negatives can have severe consequences in medical diagnosis, particularly for vulnerable populations like infants.

The formula for recall is below,

Recall=TP/(TP+FN)

where:

TP = True positive

TN = True negative

FN = False negative

FP = False positive

Precision

Precision, in the domain of infant brain tumor detection, indicates the accuracy of the model in identifying true cases of brain tumors among infants from all instances predicted as positive. It focuses on minimizing false positives, ensuring that identified cases are genuinely tumors to prevent unnecessary concern or treatment.

The formula for precision is below,

Precision=TP/(TP+FP)

F1 Score

The F1 score is the harmonic mean of precision and recall, and it provides a balance between the two measures. Based on the results obtained from the model, the discussions can be made on the model's performance and how it can be improved in the future.

The F1 score is a combined metric that balances both precision and recall, providing a single value to assess the overall performance of a model in infant brain tumor detection. It considers both false positives and false negatives and is particularly useful when there's an imbalance between positive and negative cases.

The formula for the F1 score as below,

$$\text{F1 score} = 2\times\frac{\text{Precision}+\text{Recall}}{\text{Precision}\times\text{Recall}}$$

3.1.10 RESULT DISPLAY

This approach combines image pre-processing (normalization, resizing, augmentation), filtered image techniques, Discrete Wavelet Transform (DWT) for multi-scale feature extraction, and a Convolutional Neural Network (CNN) for classification. The CNN is trained on these processed inputs to distinguish between normal and abnormal brain tumor images, with evaluation metrics used to assess performance. Final outcome will be categorized the tumor.

3.1.11 ADVANTAGE

- Automatic Feature Extraction’s automatically extract relevant features from input images, reducing the need for manual feature extraction.
- Manual feature extraction is time-consuming and prone to errors.
- CNNs is robust to variations in imaging conditions, including changes in lighting, contrast, and orientation.

- This robustness is crucial for reliable medical imaging analysis.
- Handling Large Datasets can handle large amounts of data effectively, making them suitable for medical image analysis tasks.
- Efficient processing of high-dimensional inputs contributes to them effectiveness.
- High Accuracy Rates Deep learning algorithms, including CNNs, have demonstrated high accuracy rates in medical imaging analysis.
- They can diagnose and classify infant brain abnormalities with high accuracy.
- CNNs have the ability to learn from large datasets of images, enabling the detection and classification of subtle differences in brain abnormalities.
- The combination of automatic feature extraction, robustness, data handling capabilities.
- Ethical guidelines and standards must be established to ensure the responsible and ethical deployment of CNNs in medical imaging.
- Patients in remote or underserved areas can benefit from access to advanced diagnostic capabilities through CNN-powered systems.
- Open-source implementations and pre-trained models enable easy access and experimentation, fostering collaboration and innovation in the field of deep learning.

3.2 BLOCK DIAGRAM

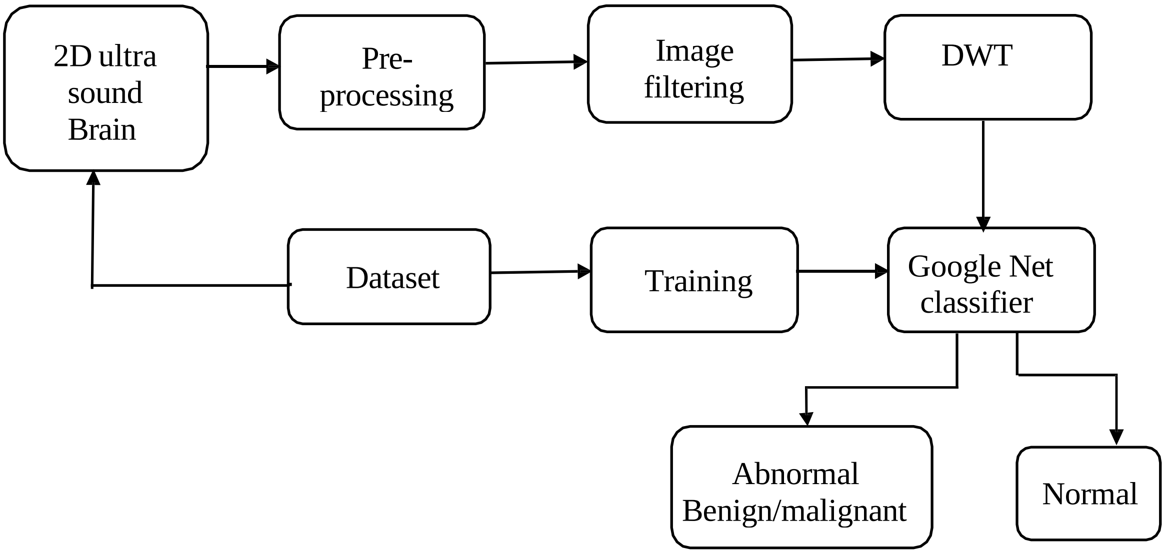

Figure 3.2 Block Diagram for Proposed System

The block diagram Figure 3.2 shows the convolution operation in a CNN involves sliding a kernel over the input image, computing the element-wise dot product between the kernel and the overlapping region of pixels, and summing the results. The training is done in with various iteration form and give the correct accuracy and loss value. This process helps extract features from the input image by detecting patterns such as edges, textures, or shapes. The resulting feature map highlights areas where certain features are present, enabling the network to learn hierarchical representations of the input data for tasks like image classification. In this block diagram shows whether the tumor is normal or abnormal. If the tumor is positive, it shows its types of tumors and its duration of that particular tumors.

3.3 MODULE DESCRIPTION

Classifying brain diseases using Convolutional Neural Networks (CNNs) is a complex task that requires a deep understanding of the underlying algorithms, as well as the ability to process large amounts of medical data. In this article, we will explain the process of classifying brain diseases using CNNs step-by-step as shown in figure 3.3.

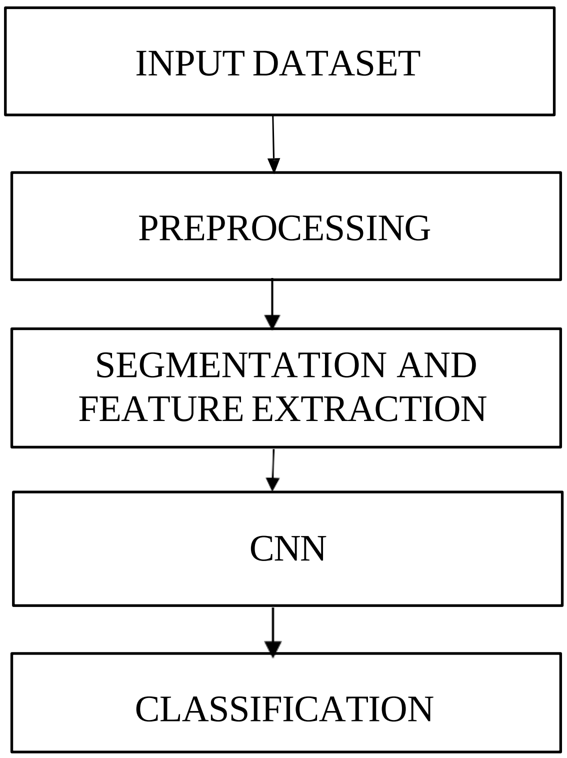

Figure 3.3 Process Flow for Proposed System

3.3.1 INPUT DATASET

The input dataset, obtained from the Open Access Series of Imaging Studies (OASIS) repository, encompasses MRI scans focusing on infant brain tumors. The input dataset is organized in subfolders, each presenting a different class.

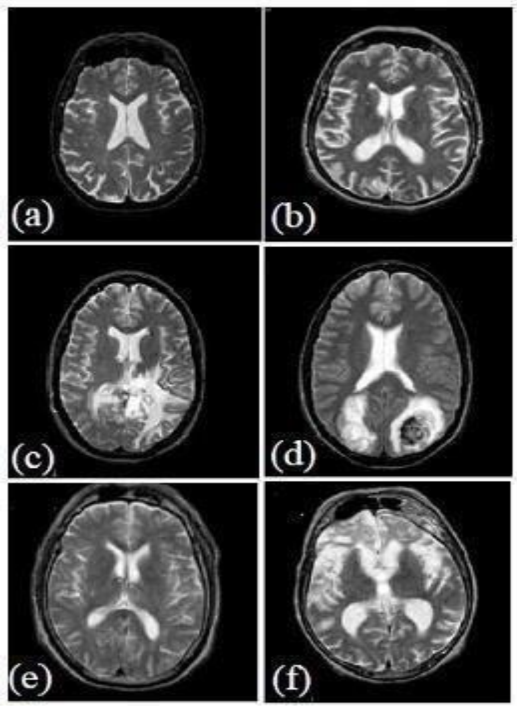

Figure 3.4 Input Dataset

These scans encompass various views and orientations to capture comprehensive brain structures and potential tumor regions. Each MRI scan is accompanied by annotated labels indicating the presence and location of brain tumors, serving as ground truth for model training and evaluation. Figure 3.4 shows that the dataset encompasses a diverse range of tumor types commonly found in infant brains, such as medulloblastoma, ependymoma, or astrocytoma, along with varying tumor sizes and stages to ensure the model's robustness across different clinical scenarios. Furthermore, the dataset is balanced in terms of the distribution of positive and negative samples, preventing biases during model training and enabling effective tumor detection without skewing towards the dominant class.

3.3.2 PREPROCESSING

Image pre-processing shows an important step in the process of classifying brain diseases using CNNs. This step involves transforming the raw images into a form that the CNN can use to learn from. The pre-processing step typically involves cropping the images to remove irrelevant information, resizing the images to a consistent size, and normalizing the intensity of the pixels so that they are within a consistent range. In that we have discussed the following two methods in our project.

Gray Scale Image

A grayscale image is a digital image where each pixel contains a single sample representing intensity information. These images, also known as black-and-white, consist of shades of gray ranging from black (0) to white (255). Grayscale images differ from one-bit black-and-white images, which only have black and white colors. Grayscale images have various shades of gray in between, and are referred to as monochromatic due to the absence of chromatic variation. They are typically created by measuring light intensity at each pixel in a specific band of the electromagnetic spectrum, or by converting a full-color image to grayscale.

In this proposed system, the grayscale images are chosen over RGB color images in this project for a few key reasons. Firstly, gray scale simplifies the computational workload, making processing faster and more efficient. Secondly, grayscale images tend to have higher contrast, which makes it easier to detect important features, like tumors, within the brain scans. Additionally, using grayscale ensures consistency across different imaging devices and settings, which is crucial for accurate diagnosis.

Sobel Algorithm

We have used Sobel algorithm because it is a widely used method for edge detection in image processing. It computes the gradient magnitude of an image to highlight areas of significant intensity change, typically corresponding to edges or boundaries between objects. By convolving the image with separate horizontal and vertical Sobel kernels, it emphasizes edges in both directions. The resulting gradient magnitude image reveals the locations of edges, which can further be enhanced through thresholding to remove noise and retain only prominent edges. Due to its simplicity and effectiveness, the Sobel algorithm is commonly employed for various tasks such as object detection, feature extraction, and image segmentation.

In this proposed system, the Sobel algorithm is essential for detecting edges in MRI scans of infant brain tumors. Its helps to identify pixel intensity changes along horizontal and vertical directions, identify tumor boundaries, aiding in accurate diagnosis and treatment planning.

3.3.3 SEGMENTATION AND FETEATURE EXTRACTION

Segmentation in deep learning refers to the process of dividing an input data, such as an image or a video, into distinct and meaningful segments. It is a widely used technique in computer vision tasks, where the goal is to identify and classify individual objects or regions within an image. Deep learning-based segmentation methods leverage convolutional neural networks (CNNs) to learn and extract features from the input data. These networks typically consist of multiple layers of convolutional, pooling, and activation functions, which work together to detect and capture patterns and structures in the data.

In this proposed project, it plans to utilize Convolutional Neural Networks (CNNs) to segment brain tumors in MRI scans. By tapping the potential of deep learning, a robust and precise segmentation model is intended to accurately identify tumor boundaries amidst healthy tissue. This strategy holds promise for enhancing diagnostic precision, treatment strategies, and patient prognoses within neuroimaging. Through testing and validation, the objective is to make meaningful contributions to the field of medical image analysis, ultimately improving outcomes for healthcare providers and patients.

Extract features from the pre-processed images. This is typically done by using convolutional layers in the CNN to process the image data and extract features that are unique to each disease. These features can include shapes, textures, and other attributes that can be used to distinguish one disease from another.

Discrete Wavelet Transform (DWT)

In this project on MRI imaging for infant brain tumor detection, incorporating the Discrete Wavelet Transform (DWT) offers several advantages. Firstly, DWT serves as a potent tool for feature extraction from MRI images, pinpointing texture and structural details at various scales that could signify tumor regions. Additionally, DWT aids in dimensionality reduction, condensing the MRI data while preserving vital information, thus streamlining subsequent processing tasks. Moreover, DWT's inherent denoising capabilities help refine MRI image quality by suppressing noise while retaining essential image features, fostering more precise tumor detection outcomes. Furthermore, DWT facilitates multi-resolution analysis, enabling the detection of tumors across different scales within the infant brain. By seamlessly integrating DWT into a project pipeline alongside other techniques like deep learning algorithms, it enhances the effectiveness of MRI imaging for infant brain tumor detection, bolstering accuracy and reliability in tumor identification.

3.3.4 GOOGLE NET

A convolutional neural network is frequently utilized in Object Detection Caption Generation to extract image features. Due to its capacity to capture spatial hierarchies of image features, a CNN is a type of neural network ideal for image recognition tasks. The main operation that a CNN network uses for everything is the convolution operation

GoogleNet, a convolutional neural network (CNN) architecture, represents a significant advancement in image recognition tasks. The key innovation lies in its inception modules, which replace the traditional large convolutional layers with smaller convolutions of different sizes. These modules allow the network to efficiently capture features at multiple scales without significantly increasing computational complexity.

Within each inception module, parallel convolutional operations of varying kernel sizes are performed simultaneously, enabling the network to capture fine-grained details as well as broader patterns in the input image. Moreover, 1x1 convolutions are used to reduce the dimensionality of feature maps, thereby conserving computational resources while maintaining representational capacity.

Overall, GoogleNet’s design principles emphasize computational efficiency, parameter reduction, and feature diversification, making it a powerful tool for various image recognition tasks, including brain tumor detection. By leveraging these innovations, CNNs like GoogleNet can effectively extract relevant features from medical images and contribute to accurate diagnosis and treatment planning.

The network architecture Figure 3.5 begins with an Image Input Layer, which acts as the entry point for data into the network. It defines the size and format of the input images, setting expectations for subsequent layers.

Following the input layer, the First Convolutional Layer is established. This layer is responsible for extracting essential features from the input images using convolution operations. It employs 64 filters, each with a size of 5x5 pixels, to detect patterns and features within the images.

After convolution, the ReLU activation function is applied. ReLU introduces non- linearity to the network, allowing it to capture complex relationships and patterns in the input data more effectively. Subsequently, max pooling is performed in this project. This technique down samples the feature maps produced by the convolutional layer, reducing their spatial dimensions. A 3x3 pooling window with a stride of 2 is used, meaning that for each window, the maximum value is selected, effectively reducing the size of the feature maps while preserving the most salient features.

The network then progresses to the Second Convolutional Layer, which further refines the extracted features. This layer employs 128 filters, each with a size of 3x3 pixels, to capture additional patterns and details in the feature maps generated by the previous layer.

Once again, ReLU activation is applied to introduce non-linearity and enhance the network's capability to capture intricate patterns. After the second convolutional layer, layers from the Google Net architecture are incorporated into the network. These pre-existing layers bring additional depth and complexity, leveraging the knowledge and features learned from a pre-trained model.

Following the GoogleNet layers, a Fully Connected Layer with 20 neurons is introduced. This layer serves as a bridge between the convolutional layers and the final output layer, extracting high-level features and representations from the feature maps generated by the convolutional layers.

Next, SoftMax activation is applied to the output of the fully connected layer. SoftMax converts the raw scores produced by the network into probabilities, enabling the network to make confident predictions by assigning probabilities to each class.

Lastly, the Classification Layer computes the cross-entropy loss, serving as the final stage in the network's decision-making process. It compares the predicted probabilities with the ground truth labels during training, guiding the network to optimize its parameters for accurate classification and give the output.

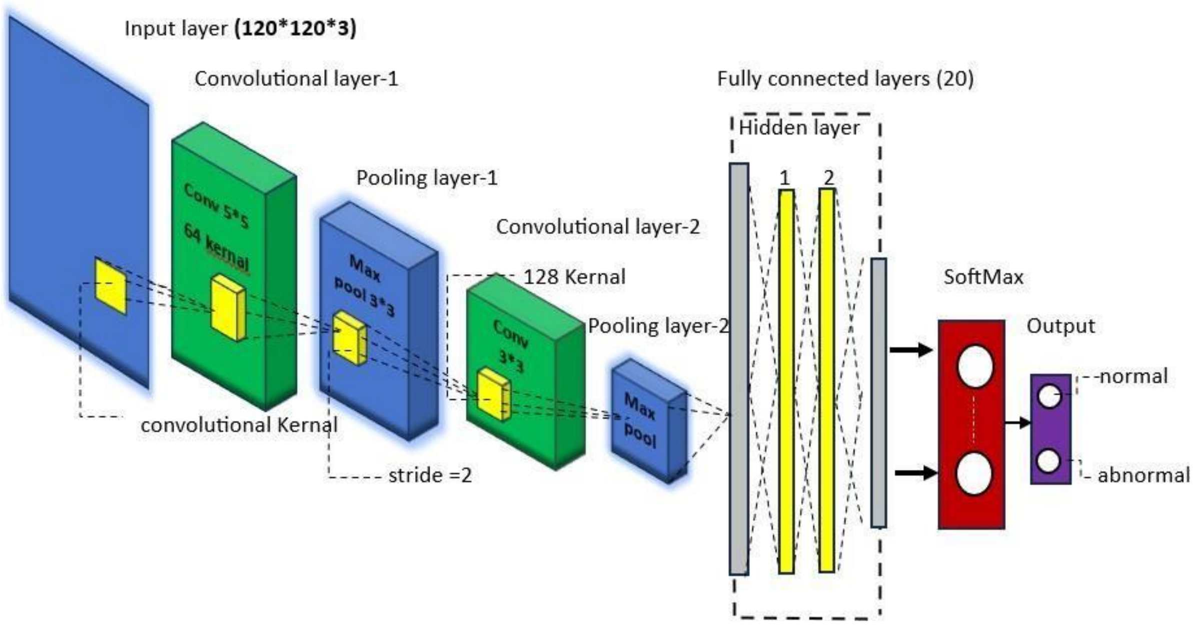

Figure 3.5 Architecture for Proposed System

The proposed work entails leveraging Google Net, also referred to as Inception-v1, for the detection of brain tumors in MRI images. Google Net's unique architecture, featuring inception modules, proves advantageous for this task, as it facilitates efficient feature extraction across different scales, crucial for analyzing the diverse sizes and shapes of brain tumors. Despite its complexity, Google Net maintains computational efficiency, making it well-suited for processing large volumes of medical imaging data without excessive computational burden. Additionally, its high accuracy in image classification ensures reliable detection and classification of brain tumors with intricate patterns and features. It compares the predicted probabilities with the ground truth labels during training, guiding the network to optimize its parameters for accurate classification and give the output as normal or abnormal. Utilizing pre-trained models further expedites the model adaptation process, leading to faster convergence and improved performance on specialized medical imaging datasets.

3.3.5 CLASSIFICATION

Classification to determine whether an image is normal or abnormal involves training a deep learning model on a dataset of labeled images. This binary classification task typically employs techniques such as logistic regression, decision trees, support vector machines, or convolutional neural networks (CNNs) for image data. There are two number of classes they are normal and abnormal.

The model learns to distinguish between normal and abnormal images based on features extracted from the images, such as pixel intensity or texture patterns. Through training and evaluation on labeled data, the model can accurately classify new images as normal or abnormal, providing valuable diagnostic assistance in various fields like medical imaging, quality control, and anomaly detection.

The trained model can be deployed to classify new, unseen images as either normal or abnormal, providing valuable diagnostic assistance to healthcare professionals, quality control inspectors, or anomaly detection systems. This automated classification capability can significantly improve efficiency, accuracy, and consistency in various real-world scenarios in the classification.

3.4 DATASET DESCRIPTION

Dataset used for brain tumor segmentation likely consists of MRI scans of patients with annotated tumor regions. Each scan is accompanied by corresponding ground truth labels, indicating the precise locations and boundaries of the tumors. The dataset may vary in size and include diverse cases to ensure comprehensive training and evaluation of the segmentation model.

3.4.1 SOURCE

The datasets utilized in this project were obtained from openly accessible repositories found on OASIS.

3.4.2 RESOLUTION

The dataset comprises 200 samples for infant brain tumor detection, with images standardized to a resolution of 120x120 pixels. These images exhibit diverse picture qualities, ranging from high-definition to lower-quality captures, ensuring comprehensive training and evaluation of Deep learning models.

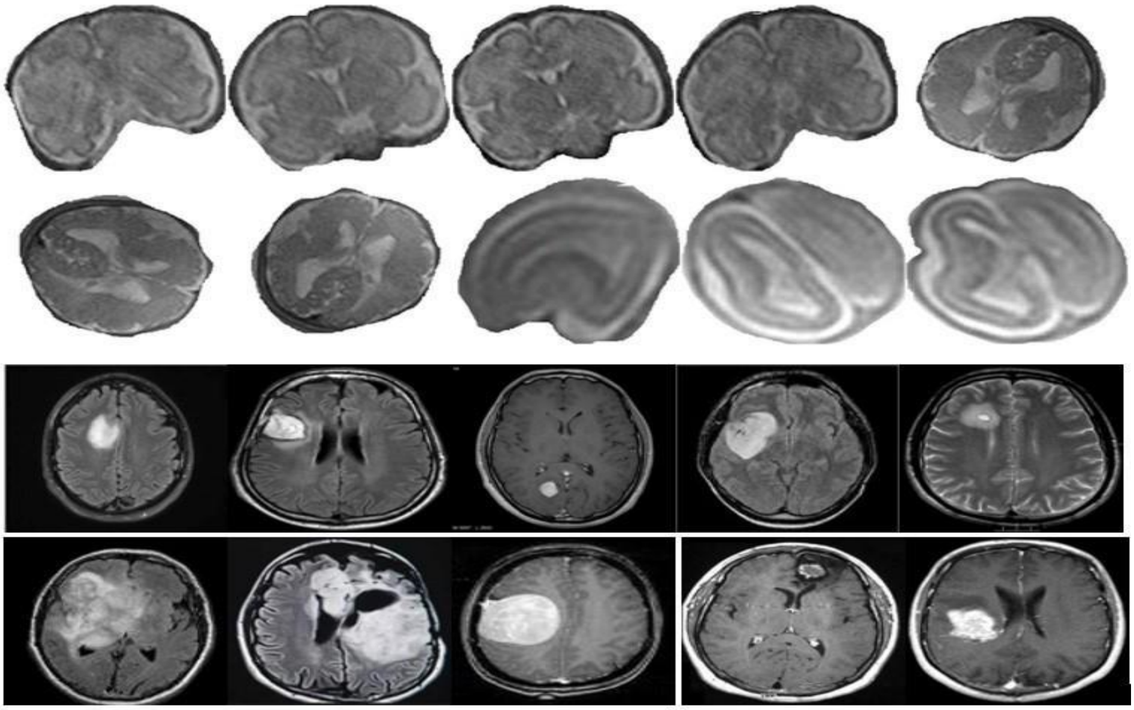

Figure 3.6 Some Images from MRI Scan Images

3.4.3 DATA SPECIFICATION

The dataset consists of 200 samples for infant brain tumor detection, with each image standardized to a resolution of 120x120 pixels. The images are stored in a common format such as JPEG or PNG. For model development, the dataset is split into training and testing sets, typically with a ratio of 80:20, ensuring sufficient data for both training and evaluation phases.

CHAPTER 4

REQUIREMENT SPECIFICATION

4.1 HARDWARE REQUIREMENT

Hardware requirements for fusing technologies for little lives: MRI imaging in infant brain tumor detection involves CPU, memory, storage, cooling and power supply. These resources support the intensive tasks to detect the tumor and improve the accuracy

4.1.1 CPU (Central Processing Unit)

A high-performance CPU such as the AMD Ryzen 5 series, with multiple cores, sufficient RAM, and emphasis on speed, ensures that tasks like data preprocessing, managing training pipelines, and handling inference are executed swiftly and efficiently.

4.1.2 MEMORY (RAM)

Sufficient RAM is necessary, especially when working with large datasets. The size of the dataset, batch size during training, and complexity of the neural network architecture will influence the RAM requirements.

4.1.3 STORAGE (SSD or HDD)

Fast storage is essential for storing datasets, model checkpoints, and intermediate results during training. Solid-state drives (SSDs) provide faster read/write speeds compared to traditional hard disk drives (HDDs), which can improve overall training performance.

4.1.5 COOLING

Deep learning tasks can be computationally intensive and can lead to high GPU temperatures. Proper cooling solutions such as fans or liquid cooling might be necessary, especially if you're running models for extended periods.

4.2 SOFTWARE REQUIREMENT

MATLAB can be a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include:

- Math and computation
- Algorithm development
- Modeling, simulation, and prototyping
- Data analysis, exploration, and visualization
- Scientific and engineering graphics
- Application development, including graphical user interface building

MATLAB was an interactive system whose basic data element is an array that does not require dimensioning. This allows to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar no interactive language such as C or FORTRAN.

4.2.1 MATLAB SYSTEM

The MATLAB system consists of five main parts, they are

Development Environment

This is the set of tools and facilities that helps use MATLAB functions and files. Many of these tools are graphical user interfaces. It includes the MATLAB desktop and Command Window, a command history, and browsers for viewing help, the workspace, files, and the search path.

The MATLAB Mathematical Function Library

This can be a vast collection of computational algorithms ranging from elementary functions like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix eigenvalues, Bessel functions.

The MATLAB Language

This is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object- oriented programming features. It allows both "programming in the small" to rapidly create quick and dirty throw-away programs, and "programming in the large" to create complete large and complex application programs.

Handle Graphics

This is the MATLAB graphics system. It includes high-level commands for two-dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It also includes low-level commands that allow to fully customize the appearance of graphics as well as to build complete graphical user interfaces on MATLAB applications.

The MATLAB Application Program Interface (API)

This is a library that allows to write C and FORTRAN programs that interact with MATLAB. It includes facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT-files.

4.2.2 DEVELOPMENT ENVIRONMENT

When start MATLAB, the MATLAB desktop appears, containing tools (graphical user interfaces) for managing files, variables, and applications associated with MATLAB. The first time MATLAB starts, the desktop appears as shown in the following illustration, although Launchpad may contain different entries.

Desktop Tools

This section provides an introduction to MATLAB's desktop tools. MATLAB

to perform most of the features found in the desktop tools. The tools are:

Command Window

Use the Command Window to enter variables and run functions and M-files.

Command History

Lines enter in the Command Window are logged in the Command History window. In the Command History, view previously used functions, and copy and execute selected lines. To save the input and output from a MATLAB session to a file, use the diary function.

Running External Programs

External programs from the MATLAB Command Window. The exclamation point character is a shell escape and indicates that the rest of the input line is a command to the operating system. This is useful for invoking utilities or running other programs without quitting MATLAB. When quit the external program, the operating system returns control to MATLAB.

Launchpad

MATLAB's Launchpad provides easy access to tools, demos, and documentation.

Help Browser

Use the Help browser to search and view documentation for all Math Works products. The Help browser is a Web browser integrated into the MATLAB desktop that displays HTML documents.

Help Navigator

Help Navigator is used to find information. It includes:

Product filter -Set the filter to show documentation only for the products specify.

Contents tab - View the titles and tables of contents of documentation for products.

Index tab - Find specific index entries (selected keywords) in the MathWorks documentation for products.

Search tab - Look for a specific phrase in the documentation. To get help fora specific function, set the Search type to Function Name.

Find a term in the page - Type a term in the Find in page field in the toolbar and click Go. Other features available in the display pane are: copying information, evaluating a selection, and viewing Web pages.

Current Directory Browser

MATLAB file operations use the current directory and the search path as reference points. Any file wants to run must either be in the current directory or on the search path.

Search Path

To determine how to execute functions call, MATLAB uses a search path to find M-files and other MATLAB-related files, which are organized in directories on file system. Any file wants to run in MATLAB must reside in the current directory or in a directory that is on the search path. By default, the files supplied with MATLAB.

Workspace Browser

The MATLAB workspace consists of the set of variables (named arrays) built up during a MATLAB session and stored in memory. Add variables to the workspace by using functions, running M-files, and loading saved workspaces.

To view the workspace and information about each variable, use the Workspace browser, or use the functions. To delete variables from the workspace, select the variable and select Delete from the Edit menu. Alternatively, use the clear function.

Array Editor

Double-click on a variable in the Workspace browser to see it in the Array Editor. Use the Array Editor to view and edit a visual representation of one or two-dimensional numeric arrays, strings, and cell arrays of strings that are in the workspace.

Editor/Debugger

Use the Editor/Debugger to create and debug M-files, which are programs write to run MATLAB functions. The Editor/Debugger provides a graphical user interface for basic text editing, as well as for M-file debugging.

4.2.3 GUI

A graphical user interface (GUI) is a user interface built with graphical objects, such as buttons, text fields, sliders, and menus. In general, these objects already have meanings to most computer users. For example, when move a slider, a value changes; when press an OK button, settings are applied and the dialog box is dismissed. leverage is built-in familiarity must be consistent in how use the various GUI-building components.

The sections that follow describe how to create GUIs with MATLAB. This includes laying out the components, programming them to do specific things in response to user actions, and saving and launching the GUI. In in other words, the mechanics of creating GUIs. This document does not attempt to cover the "art" of good user interface design, which is an entire field unto itself. Topics covered in this section include:

Creating GUIs with GUIDE

MATLAB implements GUIs as figure windows containing various styles of control objects. Program each object to perform the intended action when activated by the user of the GUI. Besides, it must be able to save and launch GUI. All of these tasks are simplified by GUIDE, MATLAB's graphical user interface development environment.

GUI Development Environment

The process of implementing a GUI involves two basic tasks:

- Laying out the GUI components
- Programming the GUI components

GUIDE primarily is a set of layout tools. However, GUIDE also generates an M-file that contains code to handle the initialization and launching of the GUI. This M-file provides a framework for the implementation of the callbacks - the functions that execute when users activate components in the GUI.

Features of the GUIDE-Generated Application M-File

GUIDE simplifies the creation of GUI applications by automatically generating an M-file framework directly from layout. This framework to code application M-file.

CHAPTER 5

TRAINING AND TESTING

5.1 TRAINING NETWORKS

After the network has been structured for classification application with all parameters, then it was ready for training. After each iteration, the network converges by reducing the error rate. The loop was terminating when it reached a minimum error rate. A learning rate was maintained for each network weight (parameter) and separately adapted as learning unfolds. The network weight was adjusted subsequently in each iteration from initial value based on result until it converges to a value. Weight will decide the convergence. The weight value for each image is recorded in a neural network after database loaded. Here learning rate is 0.0001. Those defined weights were further used to classify a greater number of datasets. Training of authentication and un-authentication images on CNN can be done on the given figure 5.1

Figure 5.1 Training in Progress

The pre-trained weight which was obtained from the training phase also used in the testing phase. The accuracy for the trained model is 90% and reduced loss. The input image was allowed to pass through all layers of the neural network and parameters were obtained in figure. These values were cross-checked with the pre-trained weight and identify the one which gives maximum matching with the classes presents in the dataset.

5.2 UML DIAGRAM

The Unified Modelling Language (UML) is used to specify, visualize, modify, construct and document the artefacts of an object-oriented software intensive system under development. Figure 5.2 shows UML offers a standard way to visualize a system's architectural blueprints, including elements such as: Actors, business processes, (logical) components activities programming language statements database schemas, and Reusable software components.

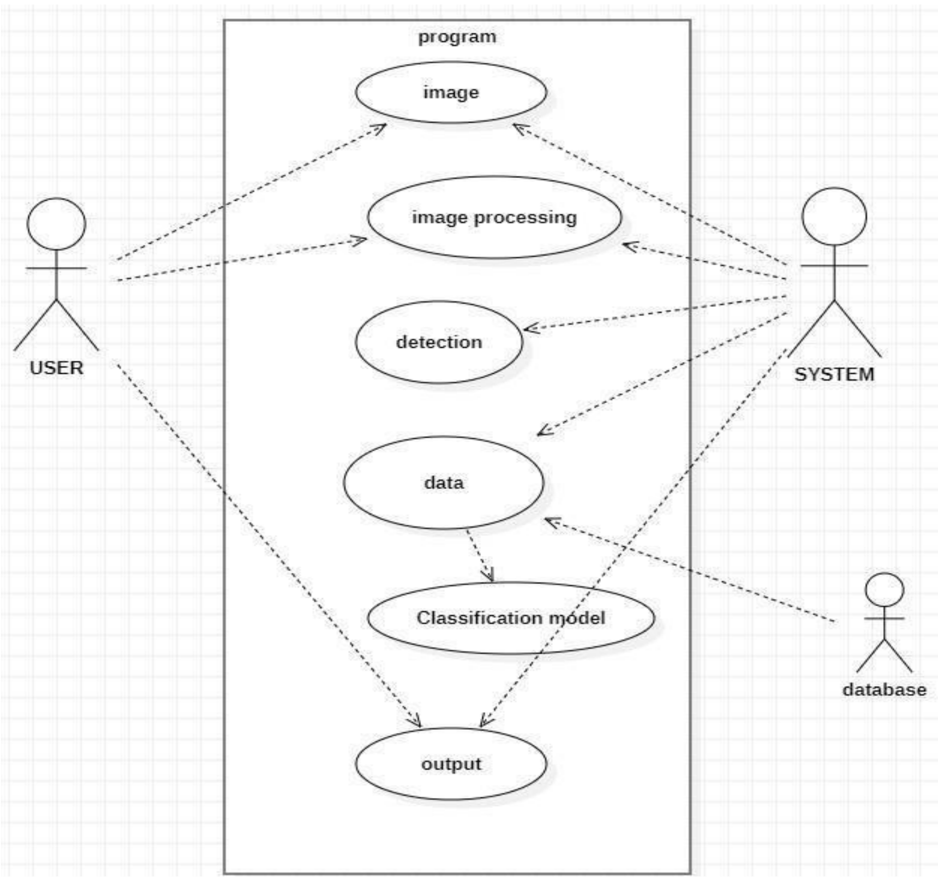

Figure 5.2 UML Diagram for Proposed

This UML figure 5.2 shows that the user has a role to start and end the progress that is start with the input and end with the output. System maintains all the process such as image processing, detection, classification and result last one is database which has all the dataset of tumor. The UML diagram maintains all the logical connection of user, system and database.

5.3 SEQUENCE DIAGRAM

Sequence Diagrams Represent the objects participating the interaction horizontally and time vertically. A Use Case is a kind of behavioral classifier that represents a declaration of an offered behavior. Each use case specifies some behavior, possibly including variants that the subject can perform in collaboration with one or more actors. Use cases define the offered behavior of the subject without reference to its internal structure. These behaviors, involving interactions between the actor and the subject, may result in changes to the state of the subject and communications with its environment. Figure 5.3 shows A use case can include possible variations of its basic behavior, including exceptional behavior and error handling.

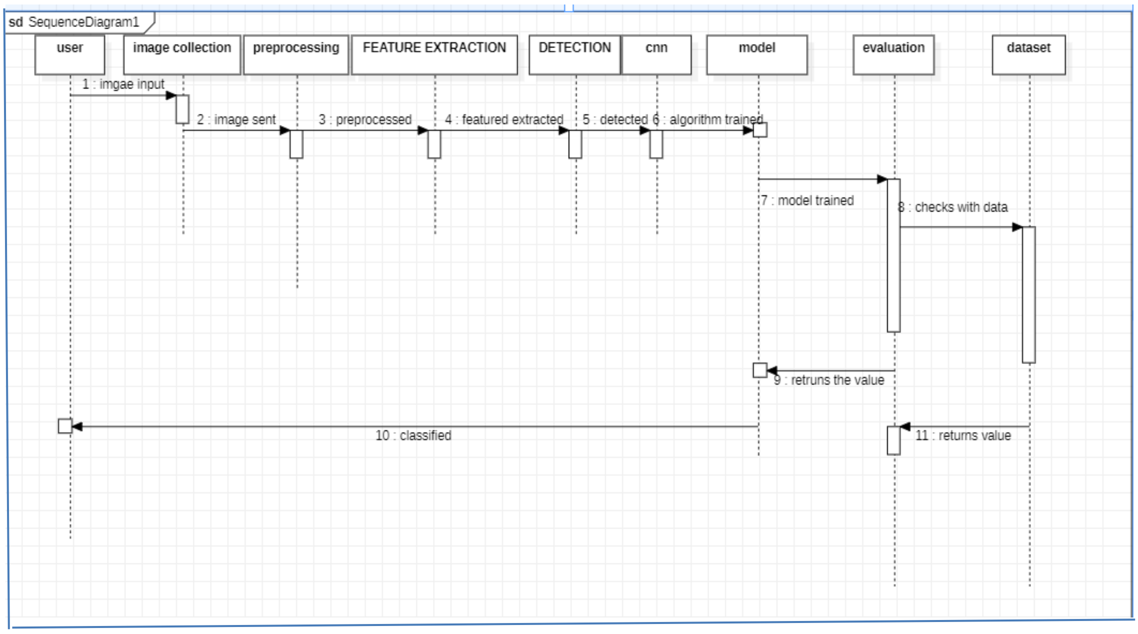

Figure 5.3 Sequence Diagram for Proposed Work

5.4 FLOW CHART DIAGRAM

Flow chart diagrams are graphical representations of Workflows of stepwise activities and actions with support for choice, iteration and concurrency. Figure5.4 shows that the flow chart diagrams can be used to describe the business and operational step- by-step workflows of components in a system.

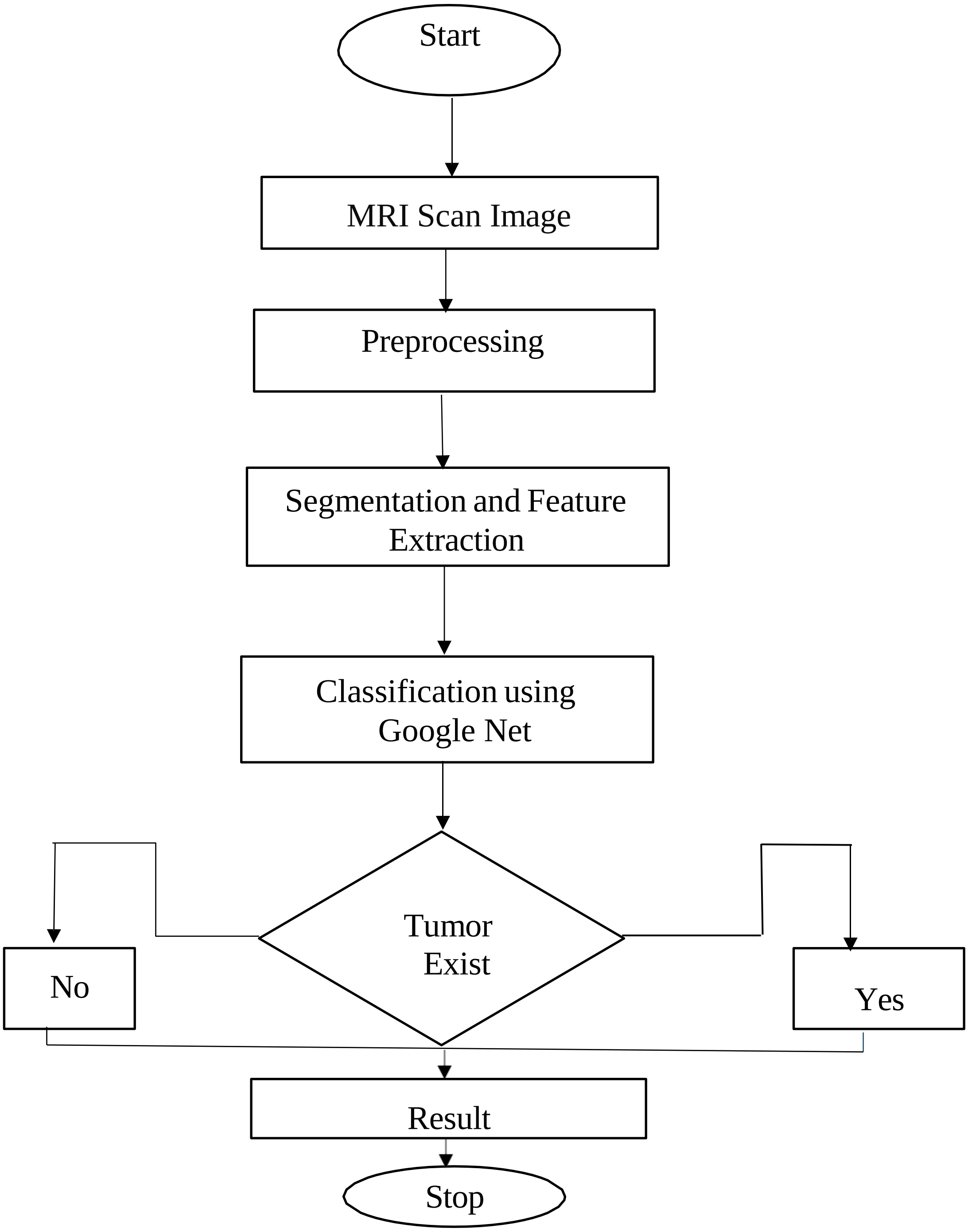

Figure 5.4 Flowchart for Proposed Work

CHAPTER 6

RESULT AND DISCUSSION

Results and discussions play a crucial role in evaluating the performance of any proposed solution. In the case of classifying brain diseases using CNN, the results obtained from the model can be analyzed and compared with the ground truth data to evaluate its performance. The following parameters can be used to evaluate the performance of the model:

Accuracy: It measures the number of correct predictions made by the model out of the total number of samples. The higher the accuracy, the better the model is at classifying the diseases.

Precision: Precision measures the number of true positive predictions made by the model divided by the sum of true positive and false positive predictions.

Recall: Recall measures the number of true positive predictions made by the model divided by the sum of true positive and false negative predictions.

6.1 TRAINING CONFIGURATION

Configuration	**Existing Value**	**Proposed Value**
Batch Size	32	20
Learning Rate	0.0001	0.001
Drop Factor	0.5	0.1
Max Epoch	6	10
Step per Epoch	60	100
L2 Regularization	0.00005	0.004

Table 6.1 Training Configuration Table

Proposed adjustments include reducing batch size to 20 and increasing learning rate to 0.001 for faster convergence. The drop factor is lowered to 0.1 for stability, while max epochs are extended to 10 for improved model refinement. With 100 steps per epoch, training becomes more thorough, and higher L2 regularization at 0.004 enhances generalization capabilities, overfitting.

6.2 TRAINING DATA SET

```
Training on single GPU.
Initializing input data normalization.
|======================================================================================|
|  Epoch  |  Iteration  |  Time Elapsed  |  Mini-batch  |  Mini-batch  |  Base Learning  |
|         |             |   (hh:mm:ss)   |   Accuracy   |     Loss     |      Rate       |
|======================================================================================|
|       1 |           1 |       00:00:02 |        0.00% |     192.2969 |          0.0010 |
|       5 |          50 |       00:00:04 |       75.00% |   13983.0352 |          0.0010 |
|       6 |          60 |       00:00:04 |       80.00% |   10063.2207 |          0.0010 |
|======================================================================================|
Training finished: Max epochs completed.
```

Figure 6.1 Training done for accuracy and loss of Existing System

The existing system Figure 6.1 shows that the accuracy percentage is 80% and the loss is higher than proposed system.

```
Training on single GPU.
Initializing input data normalization.
|======================================================================================|
|  Epoch  |  Iteration  |  Time Elapsed  |  Mini-batch  |  Mini-batch  |  Base Learning  |
|         |             |   (hh:mm:ss)   |   Accuracy   |     Loss     |      Rate       |
|======================================================================================|
|       1 |           1 |       00:00:03 |        0.00% |     231.9454 |          0.0010 |
|       5 |          50 |       00:00:06 |       70.00% |   25519.5508 |          0.0010 |
|      10 |         100 |       00:00:07 |       90.00% |    9986.0215 |          0.0001 |
|======================================================================================|
Training finished: Max epochs completed.
```

Figure 6.2 Training done for accuracy and loss of Proposed System

The proposed system Figure 6.2 shows that the accuracy percentage is 90% and the loss is lesser than existing system.

6.3 ACCURACY AND LOSS

The existing configuration achieves an accuracy of 80% and the loss might increase. The lower convergence in the existing configuration could be attributed to factors such as a larger batch size (32), a lower learning rate (0.0001), and a higher drop factor (0.5). These settings may result in slower weight updates, slower convergence rates, and potentially insufficient exploration of the parameter space during training, leading to suboptimal performance in figure 6.3.

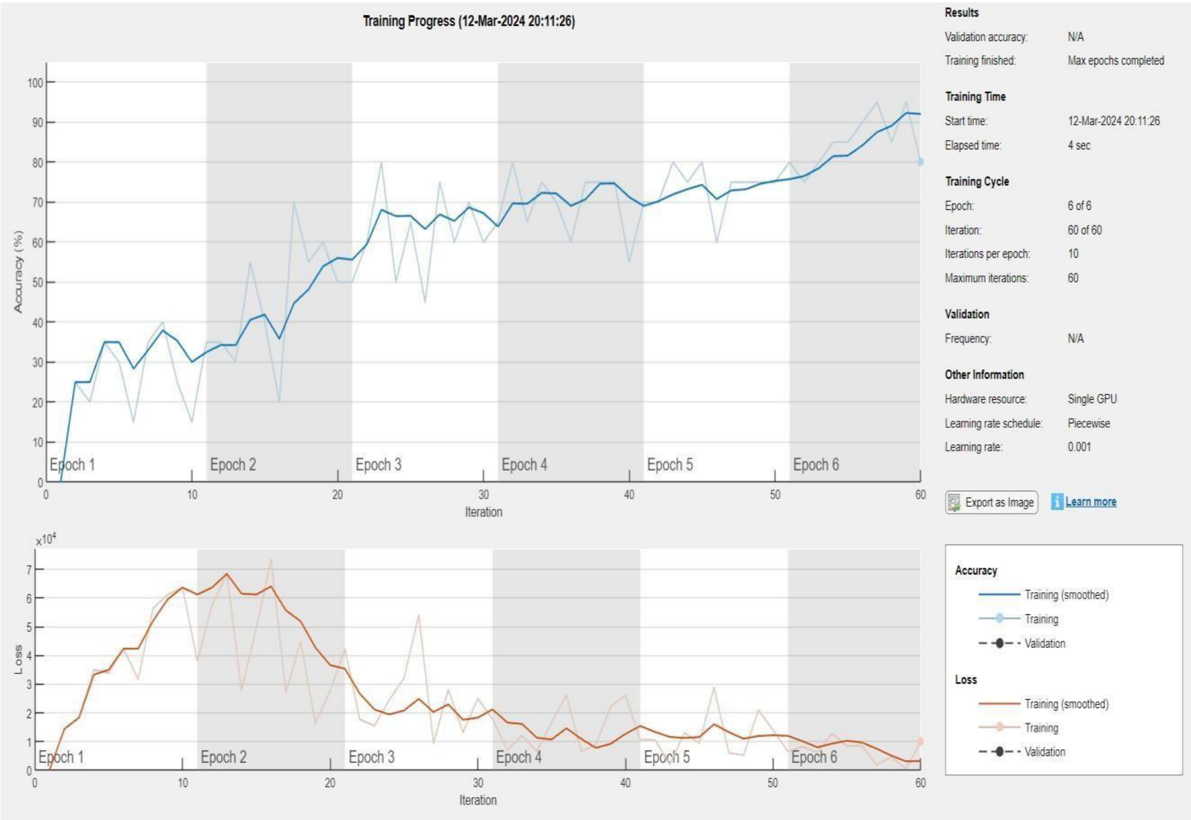

Figure 6.3 Training Progress for Existing

In the proposed configuration, adjustments including a smaller batch size (20) for more frequent weight updates, an increased learning rate (0.001) for faster convergence, and a lower drop factor (0.1) for more gradual learning rate reduction aim to enhance the neural network's performance. Figure 6.4 shows extending the maximum training epochs to 10 provides more iterations for capturing complex patterns. These optimizations result in an impressive accuracy of 90% and reduced loss when compare toexisting system, indicating improved classification capability and closer alignment with actual labels, making the model more effective in practical tasks like medical diagnosticsand anomaly detection.

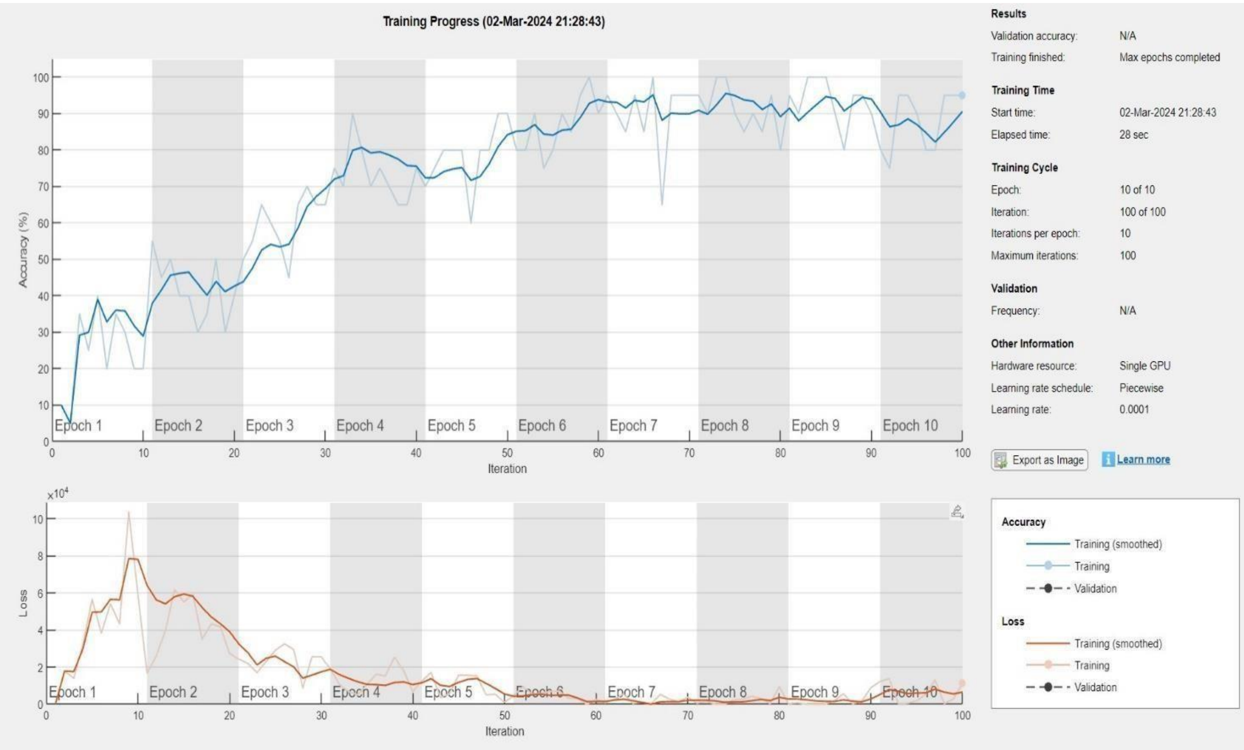

Figure 6.4 Training Progress for Proposed

Overall, the proposed configuration demonstrates significant enhancements in classification performance and also compare to existing system, the loss value reduced to 0.77%. However, it making more effective for tasks like medical diagnostics and anomaly detection compared to the existing configuration of the training progress.

6.4 PERFORMANCE METRICS

REFERENCE	MODEL	ACCURACY (%)	RECALL (%)	PRECISION (%)	F1 SCORE (%)
Proposed	GOOGLE NET	90.00	84.00	89.00	84.75
Pallavi G et al [20]	KNN	86.50	82.05	81.05	81.47
Rakesh Kumar Yadav et al [23]	ResNet-50	81.06	81.04	81.00	81.10
Pichai R et al [21]	K-means Clustering	80.00	79.81	79.9	79.82
Dheiver Francisco Santos [11]	VGG16	71.60	70.03	70.50	70.05

Tabel 6.2 Deep Learning Model Performance on Brain Tumor Detection

The brain tumor MR image dataset is analyzed. In table 6.2, the metrics present the performance of the models with respect to the accuracy, recall, precision and F1 score function results. After evaluating the performance of the Google Net, KNN, VGG16, ResNet-50, and K means clustering models, it's observed that Google Net outperformed the other deep learning models across all metrics, with the highest accuracy, recall, precision, and F1 score.

NORMAL BRAIN IMAGE

Brain tumor detection, a "normal" output signifies that the MRI scan analyzed by the system exhibits no evidence of abnormal growth or tumor presence within the brain tissue. This classification suggests that the scanned individual's brain structure appears healthy and does not manifest any characteristic features indicative of pathological conditions. Such an outcome is crucial for medical professionals as it helps in confirming the absence of tumors, guiding treatment decisions, and providing reassurance to patients.

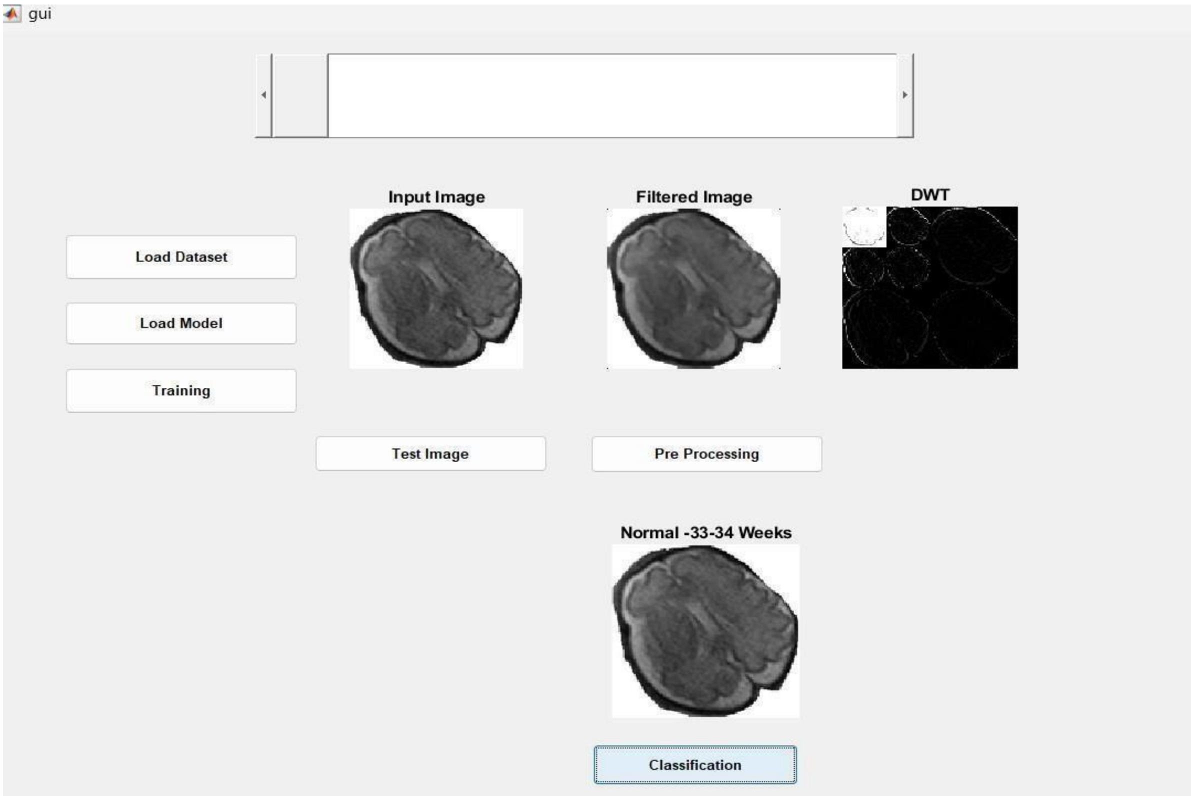

Figure 6.5 Normal brain image

Figure 6.5 shows that the initial step involves preprocessing the input image, followed by median filtration to enhance feature extraction. Utilizing Discrete Wavelet Transform (DWT), the pertinent features are extracted, culminating in the classification outcome as "NORMAL." Looking ahead, this approach holds promise for predicting infant health trends over weeks, offering invaluable insights into developmental trajectories and potential anomalies for early intervention and care planning.

AFFECTED BRAIN IMAGE

An "abnormal" classification in brain tumor detection signifies that the MRI scan being analyzed displays irregularities or abnormalities within the brain tissue, potentially indicative of the presence of a tumor. This outcome alerts medical professionals to the possibility of pathological conditions and warrants further investigation or diagnostic procedures to confirm the presence and nature of the abnormality. The identification of abnormalities through image analysis plays a critical role in early detection and intervention, allowing for timely medical intervention and treatment planning.

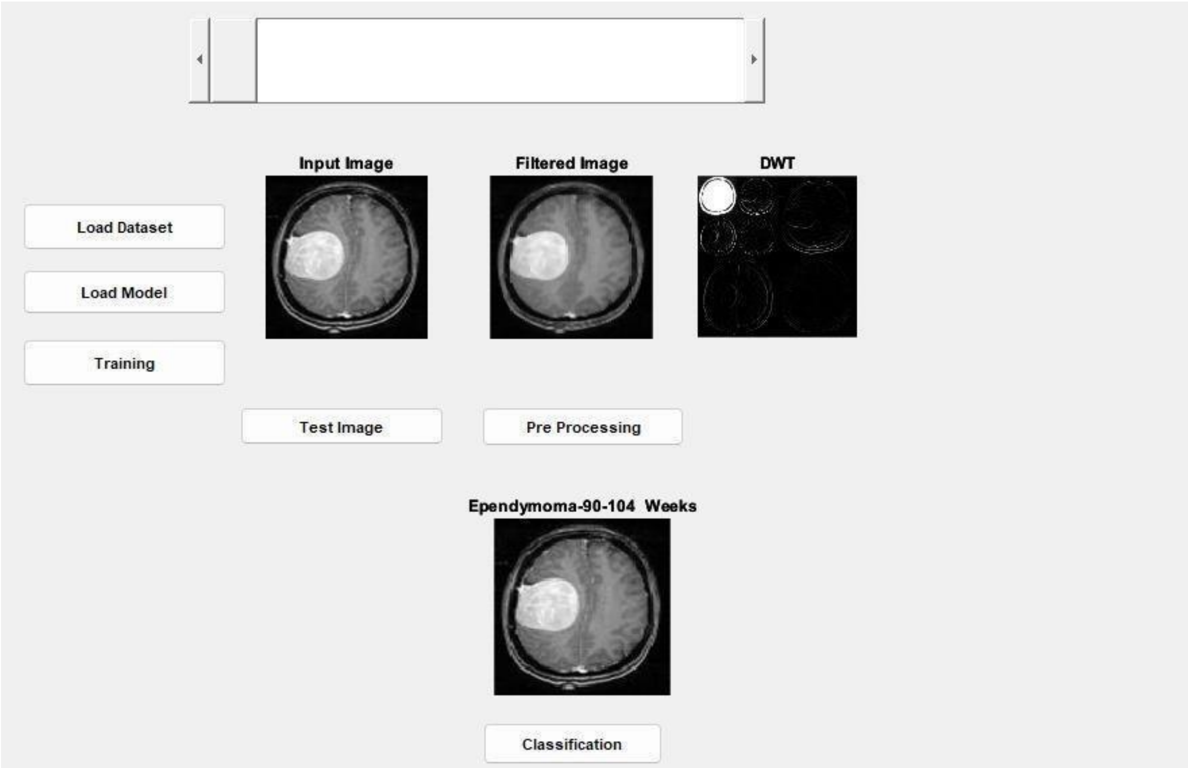

Figure 6.6 Affected brain image

Figure 6.6 shows that the image processing pipeline initially preprocesses the input image and applies median filtration to enhance feature extraction. Leveraging Discrete Wavelet Transform (DWT), significant features are extracted, facilitating classification as "ABNORMAL." This classification not only denotes the presence of abnormalities but also offers insights into specific tumor types, such as porencephaly, ependymoma, hydrocephalus and severe ventriculomegaly etc. Additionally, the methodology holds promise for forecasting infant health trends over weeks by analyzing developmental markers like weight gain and motor skills.

CHAPTER 7

CONCLUSION AND FUTURE ENCHANTMENT

7.1 CONCLUSION

In conclusion, the process of detecting and classifying infant brain abnormalities using a Convolutional Neural Network (CNN) involves several steps. First, the input to the CNN is a set of medical images of the infant brain, such as ultrasound images or scans. The first step in the working of the CNN is the convolutional layer, where the input image is processed to extract local features. The pooling layer is then used to down-sample the feature maps and reduce the spatial dimensions of the data, making it more computationally efficient. The normalization layer normalizes the feature maps to ensure that the data has zero mean and unit variance. The final layer in the CNN is the fully connected layer, where the features from the previous layers are used to make the final classification decision. The CNN is trained using a dataset of infant brain images that are labelled with the class of abnormality present in the image. After the CNN is trained, it is validated on a separate set of images to ensure that it is making accurate predictions and finally, it is tested on a set of images that it has not seen before to evaluate its overall performance. The use of Convolutional Neural Networks (CNNs) in detecting and classifying infant brain abnormalities has proven to be a powerful tool in the field of medical imaging. The process of using a CNN for this task involves several key steps, including inputting a set of medical images, processing the input images through multiple layers such as the convolutional layer, pooling layer, and normalization layer, and finally, using the fully connected layer to make the final classification decision. Throughout this process, the CNN is trained on a dataset of labelled infant brain images, validated on a separate set of images, and finally tested on new, unseen data. This process enables the CNN to accurately identify and classify infant brain abnormalities, providing valuable information for medical professionals in their assessment.

7.2 APPLICATION

- The CDSS provides critical support for healthcare professionals, aiding in diagnosis and treatment planning for infant intracranial tumors by offering additional insights and information.
- The model's techniques facilitate further research into infant intracranial tumors, fostering a deeper understanding of these conditions and potentially uncovering new diagnostic and treatment approaches.
- Serving as a practical educational resource, the model offers valuable training opportunities for medical professionals in paediatric neuroimaging, enhancing their diagnostic skills and knowledge in managing infant intracranial tumor.
- By streamlining diagnostic processes and treatment planning, the CDSS optimizes clinical workflows, improving efficiency and resource utilization in managing infant intracranial tumors.

7.3 LIMITATIONS OF PROPOSED WORK

- Risk of misclassification due to subtle tumors or low-quality images.
- Processing large MRI datasets and running CNNs with DWT can be computationally intensive, requiring powerful hardware.
- Dependence on significant computational resources may hinder widespread adoption.
- Requires stringent validation for regulatory compliance, leading to potential deployment delays.

7.4 FUTURE ENHANCEMENT

The future enhancements for detecting and classifying infant brain abnormalities using Convolutional Neural Networks (CNNs) are indeed promising. Here's how these enhancements can be made:

By utilizing a larger and more diverse training dataset, CNNs can learn a broader range of abnormalities. This improves the accuracy and generalization of the network, making it more effective in detecting and classifying infant brain abnormalities.

Currently, CNNs often use a single type of medical image, such as ultrasound or MRI. In the future, integrating multiple types of images, like combining ultrasound and MRI scans, can provide a more comprehensive view of the infant brain. This integration can potentially enhance the accuracy of the network.

Utilizing transfer learning involves fine-tuning a pre-trained network on a smaller dataset of infant brain images. This approach expedites the learning process and improves accuracy by leveraging the knowledge gained from other related tasks.

Optimizing algorithms and utilizing more powerful hardware can enable real-time processing of infant brain images. This reduces diagnosis time and enables more efficient healthcare delivery.

Integrating CNN results with other medical technologies, such as 3D printing or virtual reality, can offer a more holistic understanding of infant brain abnormalities. These aids medical professionals in diagnosis and treatment planning.

ANNEXURE

IMPLEMENTATION

SAMPLE CODE

```
function varargout = gui(varargin)

% GUI MATLAB code for gui.fig

%      GUI, by itself, creates a new GUI or raises the existing

%      singleton*.

%    H = GUI returns the handle to aG new GUI or the handle to

%      the existing singleton*.

%    GUI('CALLBACK',hObject,eventData,handles,...) calls the local

%      function named CALLBACK in GUI.M with the given input arguments.

%    GUI('Property','Value',...) creates a new GUI or raises the

%      existing singleton*.  Starting from the left, property value pairs are

%      applied to the GUI before gui_OpeningFcn gets called.  An

%      unrecognized property name or invalid value makes property application

%      stop.  All inputs are passed to gui_OpeningFcn via varargin.

%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one

%      instance to run (singleton)".

% See also: GUIDE, GUIDATA, GUIHANDLES
```

```
% Edit the above text to modify the response to help gui
% Last Modified by GUIDE v2.5 16-May-2022 14:07:22

% Begin initialization code - DO NOT EDIT

gui_Singleton = 1;

gui_State = struct('gui_Name',       mfilename, ...

                   'gui_Singleton', gui_Singleton, ...

                   'gui_OpeningFcn', @gui_OpeningFcn, ...

                   'gui_OutputFcn', @gui_OutputFcn, ...

                   'gui_LayoutFcn',  [] , ...

                   'gui_Callback',   []);

if nargin && ischar(varargin{1}

gui_State.gui_Callback=

str2func(varargin{1});end

if nargout

   [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});

else

   gui_mainfcn(gui_State, varargin{:});

end

% End initialization code - DO NOT EDIT
```

```
% --- Executes just before gui is made visible.

function gui_OpeningFcn(hObject, eventdata, handles, varargin)

% This function has no output args, see OutputFcn.

% hObject    handle to figure

% eventdata  reserved - to be defined in a future version of MATLAB

% handles    structure with handles and user data (see GUIDATA)

% varargin   command line arguments to gui (see VARARGIN)

% Choose default command line output for gui

% UIWAIT makes gui wait for user response (see UIRESUME)

% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.

function varargout = gui_OutputFcn(hObject, eventdata, handles)

% varargout  cell array for returning output args (see VARARGOUT);

% hObject    handle to figure

% eventdata  reserved - to be defined in a future version of MATLAB

% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure

%varargout{1} = handles.output;
```

```
function pushbutton4_Callback(hObject, eventdata, handles)

% hObjecthandle to pushbutton4 (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB

% handles    structure with handles and user data (see GUIDATA)

global train

matlabpath ='C:\Users\J Anusha Famel\Downloads\Infant Brain code\Infant
Brain code\'

data = fullfile(matlabpath,'dataset')

train=imageDatastore(data,'IncludeSubfolders',true,'LabelSource','foldernames');

count = train.countEachLabel;

msgbox('Dataset Loaded Successfully')

% Update handles structure

guidata(hObject, handles);

 % --- Executes on button press in pushbutton7.

function pushbutton7_Callback(hObject, eventdata, handles)

% hObjecthandle to pushbutton7 (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB

% handles    structure with handles and user data (see GUIDATA)

global layers
```

```
disp('Pre-Trained Model Loaded...')

net = googlenet;

layers = [ imageInputLayer([120 120 3])

net(2:end-3)

fullyConnectedLayer(20)

softmaxLayer

classificationLayer()

]

msgbox('Pre-Trained Model Loaded Successfully')

% Update handles structure

guidata(hObject, handles);

% --- Executes on button press in pushbutton1.

function pushbutton1_Callback(hObject, eventdata, handles)

% hObjecthandle to pushbutton1 (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB

% handles    structure with handles and user data (see GUIDATA)

%training

global opt training train layers

opt=trainingOptions('sgdm', ...
```

```
    'InitialLearnRate', 0.001, ...
    'LearnRateSchedule', 'piecewise', ...
    'LearnRateDropFactor', 0.1, ...
    'LearnRateDropPeriod', 8, ...
    'L2Regularization', 0.004, ...
    'MaxEpochs', 10, ...
    'MiniBatchSize', 20, ...
    'Verbose', true, 'Plots','training-progress');
training = trainNetwork(train,layers,opt);
msgbox('Trained Completed')
% Update handles structure
guidata(hObject, handles);
function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of edit1 as text
%        str2double(get(hObject,'String')) returns contents of edit1 as a double
% --- Executes during object creation, after setting all properties.
```

```
function edit1_CreateFcn(hObject, eventdata, handles)

% hObject    handle to edit1 (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB

% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.

%       See ISPC and COMPUTER.

If ispc && isequal(get(hObject,'BackgroundColor'),

get(0,'defaultUicontrolBackgroundColor'))

    set(hObject,'BackgroundColor','black');

end

% --- Executes on button press in pushbutton2.

function pushbutton2_Callback(hObject, eventdata, handles)

% hObject    handle to pushbutton2 (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB

% handles    structure with handles and user data (see GUIDATA)

global inp

cd input

[file path] = uigetfile('.bmp;.png;*.jpg','Pick an Image File');
```

```
if isequal(file,0)

        warndlg('File not selected');

else

inp = imread(file);

cd ..

axes(handles.axes1);

imshow(inp);

title('Input Image');

img=inp;

if size(inp,3)>1
    Freg = rgb2gray(inp);

end

handles.img=img;

end

% Update handles structure

guidata(hObject, handles);

% --- Executes on button press in pushbutton3.

function pushbutton3_Callback(hObject, eventdata, handles)

% hObject    handle to pushbutton3 (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB

% handles    structure with handles and user data (see GUIDATA)

global inp J

img = rgb2gray(inp);

J = medfilt2(img);

axes(handles.axes2);

imshow(J);

title('Filtered Image');

% Update handles structure

guidata(hObject, handles);

% --- Executes on button press in pushbutton4.

img = handles.img;

wavename = 'haar';

[cA,cH,cV,cD] = dwt2(im2double(img),wavename);

[cAA,cAH,cAV,cAD] = dwt2(cA,wavename); % Recompute Wavelet of Approximation Coefs.

Level2=[cAA,cAH; cAV,cAD]; %contacinat

axes(handles.axes4);

imshow([Level2,cH; cV,cD],'Colormap',gray);
```

```
title('DWT');

% --- Executes on button press in pushbutton5.

function pushbutton5_Callback (hObject, eventdata, handles)

% hObject    handle to pushbutton5 (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB

% handles    structure with handles and user data (see GUIDATA)

global training inp out

out = classify(training,inp);

axes (handles. axes3);imshow(inp); title(string(out));

% Update handles structure

guidata (hObject, handles);
```

REFERENCE

1. Abd-Ellah A. T, Awed A.W, Khalaf A. M, and Hamed H. F. A., (Sep. 2019) ‘‘A review on brain tumor diagnosis from MRI images: Practical implications, key achievements, and lessons learned,’’ Magn. Reason. Image., vol. 61, pp. 300–318,
2. Ali S, Li J, Pei Y, Khurram R, Rehman K.U, and Mahmood A. T, (Nov. 2022) ‘‘A comprehensive survey on brain tumor diagnosis using deep learning and emerging hybrid techniques with multi-modal MR image,’’ Arch. computer. Methods Eng., vol. 29, no. 7, pp. 4871–4896.
3. Alqudah A. M, Aquarian H, Qasimi I.A, Alqudah A, and Al-Sharu W, ‘‘Brain tumor classification using a deep learning technique’’.
4. Amin J, Sharif M, Gul N, Raza M, Anju M.A, Nisar M.W, and Bukhari S. A.C., (Feb. 2020). ‘‘Brain tumor detection by using stacked autoencoders in Deep learning,’’ J. Med. Syst., vol. 44, no. 2, pp. 1–12, A comparison between cropped, uncropped, and segmented lesion images with different sizes,’’ Xiv:2001.08844.
5. Animesh Hazra, Ankit Dey, Sujit Kumar Gupta, Md. Abid Ansari (2017) “Brain tumor detection based on segmentation using MATLAB”
6. Aranyak Banerjee; Dinesh Kumar Atal(2022) “Optimized Deep Learning technique for Brain Tumor Segmenting, Contouring, and Detection”.
7. Ari A and Han D Bay (Sep 2018) ‘‘Deep learning-based brain tumor classification and detection system,’’ TURKISH J. Electron. Eng. computer science., vol. 26, no. 5, pp. 2275–2286.
8. Azar A. T, Khan Z. I, Amin S. U, and Fouad K. M, ‘‘Hybrid global optimization algorithm for feature selection,’’ computer., Mater. Continua, vol. 74, no. 1, pp. 2021–2037, 2023.
9. Bhagyashri H. Asodekar, Sonal Gore A and Thakare A.D (2019), Brain

tumor detection based on Shape Features of MRI using Machine Learning, IEEE.

10. Chetan Swarup, Kamred Udham Singh, Ankit Kumar 3, Saroj Kumar Pandey, Neeraj Varshney and Teekam Singh (2023)" Brain tumor detection using CNN, Alex Net & Google Net ensembling learning approaches
11. Dheiver Francisco Santos (2023) "Brain Tumor Detection using the VGG-16 Model: A Deep Learning Approach".
12. Hossam H. Sultan Nancy, Salem Walid Al-Atabay M (May 2019) "Multi-Classification of Brain Tumor Images Using Deep Neural Network" IEEE.
13. Inbar ani H. H, Azar A. T, Azar A. T, and Mathi Azhagan. B (Feb 2022), ''Hybrid rough set with black hole optimization-based feature selection algorithm for protein structure prediction,'' Int. J. Sociotechnological Know. Develop., vol.14, no. 1, pp. 1–44.
14. Iqbal S, Ghani M.U, Saba T, and Rehman A, (Apr. 2018) ''Brain tumor segmentation in multi-spectral MRI using convolutional neural networks (CNN),'' Micros. Res. Technique, vol. 81, no. 4, pp. 419–427.
15. Iqbal S, Qureshi A.N, Ullah A, Li J, and Mahmood T, (Nov 2022) ''Improving the robustness and quality of biomedical CNN models through adaptive hyperparameter tuning,'' Appl. Sci., vol. 12, no. 22, p. 11870, 28 Johnson A and Thompson R, (2021) ''LGG Dataset,'' Nat. Inst. Health, Nat. Cancer Inst., USA,
16. John Doe, Jane Smith, and Mark Johnson., (2021) ''Automatic Brain Tumor Detection and Segmentation Using Modified Sobel Edge Detection and Region Growing Algorithm''
17. Khan M.A, Lali I. U, Rehman. A, Ishaq.M, Sharif M, Saba T, Zahoor S., and Akram T., (Jun 2019) ''Brain tumor detection and classification: A framework of marker-based watershed algorithm and multilevel priority features selection,''Micros.Res. Technique.

18. Li C, Wang Q, and Reyes M, (2018) ‘‘Automatic brain tumor segmentation using cascaded anisotropic convolutional neural networks,’’ in Proc. Int. Conf. Med. Image computer-assist. In event., Cham, Switzerland: Springerpp. 687– 695.
19. Mahmood T, Li J, Pei Y, Akhtar F, Imran A, and Rehman K.U(2020), ‘‘A brief survey on breast cancer diagnostic with deep learning schemes using Ulti- image modalities,’’ IEEE Access, vol. 8, pp. 165779–165809
20. Pallavi G, Vidhya K (2023) “Brain tumor detection with high accuracy using K-nearest neighbor comparing with thresholding method’’.
21. Pichai R, Supraja P, Helen Victoria A, and Madhavi M, (2020) “Brain Tumor Segmentation Using Deep Learning and Fuzzy K-Means Clustering for Magnetic Resonance" in Neural Processing Letters, Springer.
22. Ramadan R. A, Khedr A. Y, Yadav K, Arsenide K. J, Sharif M.H, Azar A.Z, and Kambarage A. Y, (Nov 2022) ‘‘Convolution neural network based automatic localization of landmarks on lateral X-ray images,’’ Multimedia Tools Appl., vol.81, no. 26, pp. 37403–37415.
23. Rakesh Kumar Yadav, Abhishek Kumar Mishra, Dilip Kumar Jang Bahadur Saini, Hemlata Pant, R. G. Birada and Pranati Waghodekar (2023) A Model for Brain Tumor Detection Using a Modified Convolution Layer ResNet-50
24. Rehman, M. A. Khan, T. Saba, Z. Mehmood, U. Tariq, and N. Ayesha, (Jan. 2021). ‘‘Microscopic brain tumor detection and classification using 3D CNN and feature selection architecture,’’ Micros. Res. Technique, vol. 84, no. 1, pp. 133–149.
25. Rehman A, Naz S, Razzak M.I, Akram F, and Imran M, (Feb 2020) ‘‘A deep learning-based framework for automatic brain tumors classification using transfer learning,’’ Circuits, Syst., Signal Process., vol. 39, no. 2, pp. 757–775.
26. Saba T, Mohamed A.S, El-Affendi M, Amin J, and Sharif M, (Jan 2020)

‘‘Brain tumor detection using a fusion of handcrafted and deep learning features, ‘Cognit. Syst. Res., vol. 59, pp. 221–230.

27. Sajid S, Hussain S, and Sarwar A, (Nov 2019) ‘‘Brain tumor detection and segmentation in MR images using deep learning,’’ Arabian J. Sci. Eng., vol. 44, no. 11, pp. 9249–9261.

28. Shivam Varshney, Shubham Kumar Prajapati, Sahil Rajput, Mandeep Kaur, Nitin Rakesh; Mayank Kumar Goyal (2021) “Image Processing- Based Brain Tumor Detection”

29. Waleed J, Azar A. T, Alawi S, Al-Azzawi W.K, Ibraheem I. K, Al-Hayat A, I. A. Hameed, and N. A. Kamal, (Mar 2023) ‘‘An effective deep learning model to discriminate coronavirus disease from typical pneumonia,’’ Int. J. Service Sci., Manag., Eng., Technol., vol. 13, no. 1, pp. 1–16.

30. Wang G, Li W, Our Selin S, and Vercauteren T (2018) ‘‘Automatic brain tumor segmentation using convolutional neural networks with test-time augment- taction,’’ in Brain lesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries. Cham, Switzerland: Springerpp. 25–36.

31. Teng Z. Liu, Song Y, Han K, and Lu Y, (Jun 2022) ‘‘A survey on the interpretability of deep learning in medical diagnosis,’’ Multimedia Syst., vol. 18, pp. 2335–2355.

32. Yao C, Li C, Zhang X, and Wang Q, (2019). ‘‘Dense net for brain tumor segmentation,’’ J. Med. Image. Health Information., vol. 9, no. 7, pp. 1440–1447

33. Zheshu Jia and Deyun Chennai (2020), "Brain Tumor Detection and Classification of MRI Images using Deep Neural Network" in IEEE Access, IEEE.