A Project Report on

Bio-Medical Image Segmentation And Detection For Brain Tumour And Skin Lesions Diseases Through U-Net

A Dissertation submitted to JNTU Hyderabad in partial fulfillment of the academic requirements for the award of the degree.

Bachelor of Technology

In

Computer Science and Engineering

Submitted by

G.HARSHITHA
(20H51A0511)

G.SURESH KUMAR
(20H51A0594)

P.DEEPIKA
(20H51A05P7)

Under the esteemed guidance of

Ms. Y. Sailaja
(Assistant Professor)

Department of Computer Science and Engineering

CMR COLLEGE OF ENGINEERING & TECHNOLOGY

(UGC Autonomous)

*Approved by AICTE *Affiliated to JNTUH *NAAC Accredited with A^+ Grade

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD - 501401.

2020- 2024

CMR COLLEGE OF ENGINEERING & TECHNOLOGY

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD – 501401

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Major Project report entitled **"Bio-Medical Image Segmentation And Detection For Brain Tumour And Skin Lesions Diseases Through U-Net"** being submitted by Gorrepati Harshitha (20H51A0511), Gummadi Suresh Kumar (20H51A0594), Pucha Deepika (20H51A05P7) in partial fulfillment for the award of Bachelor of Technology in **Computer Science And Engineering** is a record of bonafide work carried out under my guidance and supervision.

The results embodies in this project report have not been submitted to any other University or Institute for the award of any Degree.

Ms. Y. Sailaja
Assistant Professor
Dept. Of CSE

Dr. Siva Skandha Sanagala
Associate Professor and HOD
HOD/CSE

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

With great pleasure we want to take this opportunity to express our heartfelt gratitude to all the people who helped in making this project a grand success.

We are grateful to **Ms. Y. Sailaja Assistant Professor** , Department of Computer Science and Engineering for his valuable technical suggestions and guidance during the execution of this project work.

We would like to thank**, Dr. Siva Skandha Sanagala,** Head of the Department of Computer Science and Engineering, CMR College of Engineering and Technology, who is the major driving forces to complete our project work successfully.

We are very grateful to **Dr. Ghanta Devadasu**, Dean-Academics, CMR College of Engineering and Technology, for his constant support and motivation in carrying out the project work successfully.

We are highly indebted to **Major Dr. V A Narayana,** Principal, CMR College of Engineering and Technology, for giving permission to carry out this project in a successful and fruitful way.

We would like to thank the **Teaching & Non- teaching** staff of Department of Computer Science and Engineering for their co-operation

We express our sincere thanks to **Shri. Ch. Gopal Reddy**, Secretary& Correspondent, CMR Group of Institutions, and Shri Ch Abhinav Reddy, CEO, CMR Group of Institutions for their continuous care and support.

Finally, we extend thanks to our parents who stood behind us at different stages of this Project. We sincerely acknowledge and thank all those who gave support directly or indirectly in completion of this project work.

G. Harshitha	20H51A0511
G. Suresh Kumar	20H51A0594
P. Deepika	20H51A05P7

TABLE OF CONTENTS

List of Figures

ABSTRACT

The research concentrates on enhancing the precise identification and localization of diseases within medical images, a pivotal component of medical imaging analysis crucial for diagnoses and treatment planning. Employing the U-Net architecture, renowned for its effectiveness in biomedical image segmentation, the study targets the detection of brain tumors and skin lesions in MRI or CT scans. Remarkably, the proposed method not only identifies diseases but also provides intricate details regarding their dimensions and spatial arrangement. Extensive experimental validation showcases the method's superiority over existing approaches, underscoring its potential for holistic learning in medical image analysis.

CHAPTER 1
INTRODUCTION

CHAPTER 1

INTRODUCTION

1.1 Problem Statement

Medical imaging technologies have significantly transformed the landscape of disease diagnosis and injury assessment in healthcare. Among the widely used modalities such as radiography, magnetic resonance imaging (MRI), ultrasound, and computed tomography (CT), interpreting the resulting images requires substantial analysis. Image segmentation emerges as a crucial technique in this regard. Through image segmentation, healthcare professionals can pinpoint regions of interest within medical images, particularly those affected by diseases. This process enables the extraction of pertinent information, including the presence of disease within the image and its precise location. Accurate segmentation plays a pivotal role in various aspects of healthcare, ranging from diagnosis and treatment planning to surgical interventions and treatment monitoring.

1.2 Research Objective

The objective of the Bio-Medical Image Segmentation System utilizing U-Net are centred around advancing the accuracy and efficiency of medical image analysis. The primary aim is to enhance the precision of identifying and delineating specific structures within biomedical images, such as tumours or lesions, through a sophisticated neural network architecture. This system addresses the growing demand for automated and reliable segmentation techniques, offering a valuable tool for healthcare professionals in tasks like disease diagnosis, treatment planning, and monitoring.

1.3 Project Scope and Limitations

Project Scope

The scope extends to various medical imaging modalities, including MRI and CT scans, emphasizing the versatility of U-Net in handling complex and diverse data. By providing detailed and accurate segmentation results, the objective is to streamline clinical workflows, improve diagnostic outcomes, and ultimately contribute to more effective and personalized patient care in the field of biomedical imaging.

Limitations

- The system's performance may be influenced by the quality of input images.
- Noisy or low-resolution images could impact the accuracy of segmentation, highlighting the importance of obtaining high-quality input for optimal results.
- Large Data Requirements: U-Net models, like many deep learning models, require large amounts of labeled data for training. Gathering and annotating medical images can be a time-consuming and expensive process, and in some cases, there may be limited access to such data due to privacy and legal restrictions.

CHAPTER 2
BACKGROUND WORK

CHAPTER 2

BACKGROUND WORK

2.1 Brain tumor segmentation based on deep learning and an attention mechanism using MRI multi-modalities brain images

2.1.1 Introduction:

Brain tumor segmentation from medical imaging plays a pivotal role in diagnosis, treatment planning, and monitoring of brain disorders. Traditional methods for segmenting brain tumors from magnetic resonance imaging (MRI) often encounter challenges such as complexity in tumor shapes, heterogeneous appearances, and varying image qualities. However, recent advancements in deep learning techniques have shown promising results in medical image segmentation tasks.

In this research paper, titled "Brain Tumor Segmentation Based on Deep Learning and an Attention Mechanism Using MRI Multi-Modalities Brain Images," Ramin Ranjbarzadeh explores the utilization of deep learning models enhanced with an attention mechanism for accurate and efficient brain tumor segmentation from multi-modal MRI images. By leveraging the complementary information offered by different MRI modalities, including T1-weighted, T2-weighted, and fluid-attenuated inversion recovery (FLAIR) images, the proposed approach aims to improve the delineation of brain tumor boundaries while minimizing false positives and false negatives.

The integration of attention mechanisms within deep learning architectures enhances the model's ability to focus on relevant regions of interest, thereby improving segmentation accuracy. Moreover, the use of multi-modal MRI images enables the model to capture diverse tumor characteristics, leading to more comprehensive segmentation results.

Ranjbarzadeh's research contributes to the growing body of literature on medical image analysis and deep learning applications in healthcare. The proposed methodology holds promise for enhancing clinical workflows by providing clinicians with reliable tools for accurate brain tumor segmentation, ultimately facilitating timely diagnosis and treatment planning for patients with brain disorders.

2.1.2 Merits, Demerits and Challenges :

Merits :

1. Innovative Approach : Ranjbarzadeh's research introduces an innovative approach by combining deep learning techniques with attention mechanisms for brain tumor segmentation. This integration allows the model to effectively focus on relevant features in multi-modal MRI images, potentially leading to more accurate segmentation results.

2. Utilization of Multi-Modal MRI Images : By utilizing multi-modal MRI images (such as T1-weighted, T2-weighted, and FLAIR images), the proposed methodology captures diverse tumor characteristics, enhancing the comprehensiveness of the segmentation process. This multi-modal approach can potentially improve the robustness of the segmentation model.

3. Clinical Relevance : Accurate segmentation of brain tumors is crucial for clinical decision-making, treatment planning, and patient management. Ranjbarzadeh's research directly addresses this clinical need by developing a segmentation method tailored for MRI-based brain tumor analysis, potentially benefiting healthcare providers and patients.

4. Potential for Automation : Deep learning-based segmentation methods have the potential to automate and expedite the segmentation process, reducing the manual effort required by clinicians. If successful, Ranjbarzadeh's approach could contribute to more efficient workflows in clinical practice.

Demerits:

1. Data Limitations : Deep learning models heavily rely on large and diverse datasets for training to generalize well to unseen data. The effectiveness of Ranjbarzadeh's approach may be limited if the dataset used for training lacks diversity or is insufficiently large, potentially leading to overfitting or limited generalizability.

2. Validation and Clinical Adoption : While the proposed methodology may demonstrate promising results in experimental settings, its validation on independent datasets and in real-world clinical settings is essential to assess its robustness and clinical utility. Without extensive validation and clinical adoption, the practical impact of the research may be limited.

3. Interpretability : Deep learning models, particularly those with complex architectures like attention mechanisms, often lack interpretability. Clinicians may require transparent and interpretable models to understand the basis of segmentation decisions and trust the results provided by the system.

4. Computational Complexity : Deep learning models, especially those incorporating attention mechanisms, can be computationally intensive and may require substantial resources for training and inference. The computational complexity of the proposed methodology could be a barrier to its widespread adoption, particularly in resource-constrained environments.

Challenges:

1. Data Availability and Quality : Obtaining high-quality and diverse multi-modal MRI datasets with accurately annotated tumor boundaries can be challenging. Limited availability of such datasets may hinder the training and evaluation of the proposed method, leading to potential biases and limitations in generalization.

2. Optimization of Hyperparameters : Tuning hyperparameters such as learning rates, dropout rates, and network architectures for optimal performance can be challenging and time-consuming. Finding the right balance between model complexity and generalization is crucial for achieving high segmentation accuracy.

2.1.3 Implementation:

The research proposes a novel method for brain tumour segmentation using deep learning and an attention mechanism applied to multi-modal MRI images. One of the key contributions is the introduction of a distance-wise attention (DWA) module, which considers the distance between the centre of the tumour and the expected area to enhance feature selection in each MRI modality slice.

The proposed cascade CNN model integrates both local and global information from different MRI modalities, including FLAIR, T1-contrasted (T1C), T1-weighted (T1), and T2-weighted (T2) images. Additionally, Z-Score normalized versions of these modalities are incorporated to improve segmentation accuracy without adding complexity to the network structure.

To address overfitting issues, a powerful preprocessing step is employed to eliminate approximately 80% of insignificant information from each input image. This reduction in the number of pixels to investigate helps in simplifying the segmentation task without requiring a complex deep neural network architecture.

The cascade CNN model consists of two routes for extracting local and global features from the input modalities. The first route focuses on detecting pixels on the tumour border (global feature extraction), while the second route is dedicated to labelling pixels inside the tumour (local feature extraction). Convolutional layers are the core building blocks of the network, responsible for extracting low-level, mid-level, and high-level features from the input data.

In the proposed architecture, a distance-wise attention mechanism is applied at the end of the network to address the overfitting problem. This mechanism leverages the distance-wise dependencies within each MRI modality slice to select relevant features for segmentation. By considering the effect of the tumour centre's location, the model can differentiate between pixels belonging to different tumour classes more effectively.

The network utilizes activation functions such as ReLU to introduce non-linearity and dropout layers to mitigate overfitting. Pooling layers are incorporated to summarize key information and reduce the dimensionality of feature maps, thereby improving training efficiency.

For training, the stochastic gradient descent approach with cross-entropy loss function is employed to handle class imbalance. The loss function calculates the discrepancy between the ground truth and the network's predicted output, facilitating multi-class classification of pixels into tumour and non-tumour classes.

Overall, the proposed method offers a robust approach to brain tumour segmentation by effectively integrating deep learning, attention mechanisms, and multi-modal MRI data. The utilization of a cascade CNN model with distance-wise attention allows for accurate segmentation while addressing challenges such as overfitting and class imbalance.

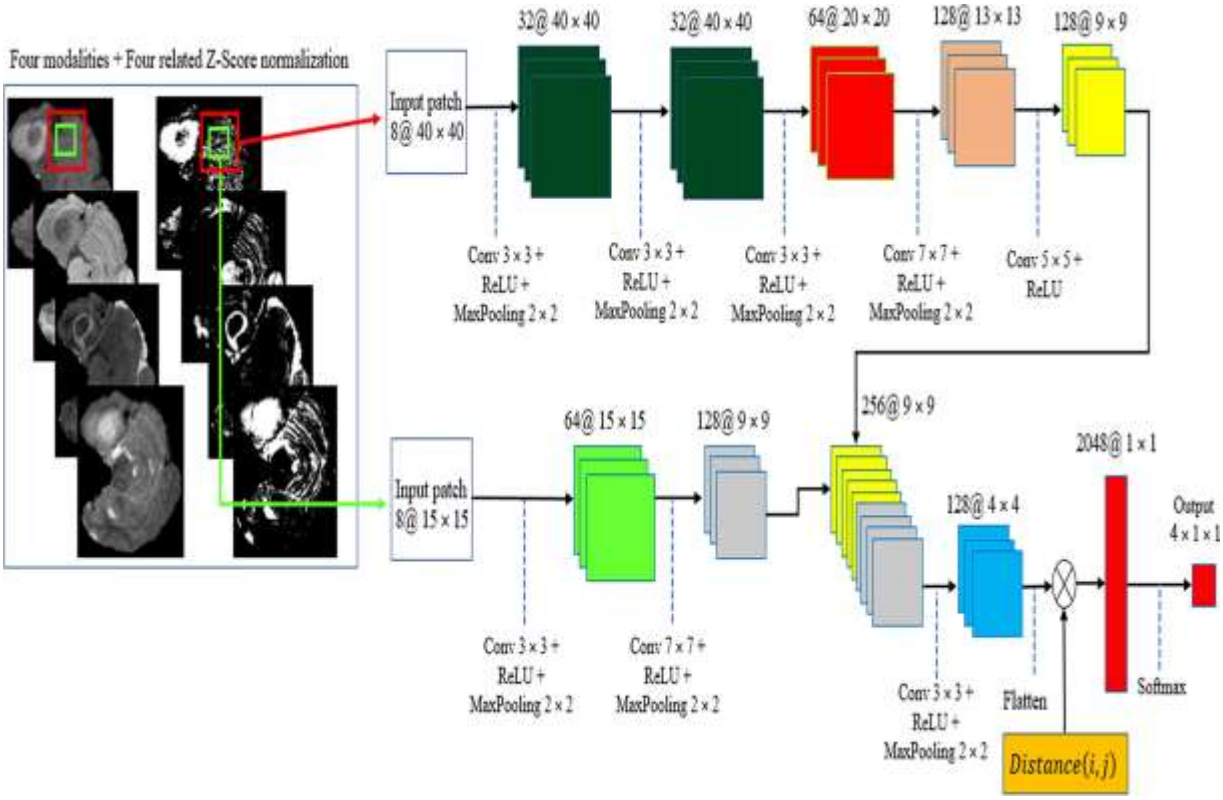

Fig* 2.1.3*: Implemented Cascade Structure

2.2 Skin Lesion Segmentation Using Deep Learning with Auxiliary Task

2.2.1 Introduction :

Skin lesion segmentation is a critical task in computer-aided diagnosis systems for dermatology, aiding in the early detection and treatment of skin diseases such as melanoma. Traditional methods for skin lesion segmentation often face challenges due to variations in lesion appearance, texture, and color, making automated segmentation a complex problem. However, recent advancements in deep learning techniques have shown promise in improving segmentation accuracy and efficiency.

In the research paper titled "Skin Lesion Segmentation Using Deep Learning with Auxiliary Task" by Lina Liu, a novel approach is proposed to address the challenges of skin lesion segmentation using deep learning methods enhanced with auxiliary tasks. The study aims to improve the accuracy and robustness of skin lesion segmentation by incorporating additional tasks into the training process.

The proposed methodology leverages deep learning architectures, such as convolutional neural networks (CNNs), to learn discriminative features from skin lesion images. Additionally, auxiliary tasks are introduced to provide supplementary information to the main segmentation task, facilitating the learning process and enhancing the model's performance.

Liu's research contributes to the growing body of literature on medical image analysis and deep learning applications in dermatology. By exploring the integration of auxiliary tasks with deep learning models, the study aims to advance the state-of-the-art in skin lesion segmentation, ultimately benefiting healthcare practitioners and patients through improved diagnostic accuracy and efficiency.

The introduction of this research paper sets the stage for discussing the methodology, experimental results, and conclusions, highlighting the significance of the proposed approach in the context of skin lesion segmentation and computer-aided diagnosis systems for dermatology.

2.2.2 Merits , Demerits and Challenges :

Merits:

1. Innovation: If the paper introduces novel techniques or methodologies for skin lesion segmentation, it would be considered a significant contribution to the field.

2. Performance Improvement: If the proposed deep learning model with auxiliary tasks demonstrates superior performance compared to existing methods in terms of segmentation accuracy, efficiency, or robustness, it would be a major merit.

3. Generalizability: If the proposed method is shown to generalize well across different datasets, skin lesion types, and imaging conditions, it would enhance the practical applicability of the research.

4. Clinical Relevance: If the segmentation results have clinical relevance and can potentially aid dermatologists in diagnosing skin lesions accurately and efficiently, the paper would be highly valued in the medical community.

5. Methodological Rigor: If the paper provides thorough experimental validation, including proper dataset selection, evaluation metrics, and comparison with state-of-the-art methods, it would strengthen the credibility of the research findings.

Demerits :

1. Overfitting: If the proposed model is prone to overfitting, especially when trained on limited or biased datasets, it would limit the generalizability of the research.

2. Lack of Validation: If the proposed method lacks validation on independent datasets or real-world clinical scenarios, its practical utility may be questioned.

3. Complexity and Efficiency Trade-off: If the proposed model is computationally expensive or resource-intensive without significant improvement in segmentation performance, it may not be practical for real-time or large-scale deployment.

4. Limited Scope: If the research focuses solely on segmentation without considering other aspects such as lesion classification or localization, it may overlook important clinical requirements.

5. Ethical Considerations: If the paper fails to address ethical considerations such as data privacy, fairness, and potential biases in the dataset, it may raise concerns regarding the ethical implications of the research.

Challenges :

1. Annotation Variability: Skin lesion annotation can be subjective and prone to inter-observer variability, leading to inconsistencies in ground truth annotations. Ensuring high-quality annotations for training deep learning models is crucial but can be challenging and time-consuming.

2. Complexity of Lesion Appearance: Skin lesions exhibit diverse appearances in terms of color, texture, shape, and size, making accurate segmentation challenging. Deep learning models must effectively capture and learn discriminative features from these complex lesion characteristics

2.2.3 Implementation :

The proposed method for skin lesion segmentation incorporates three main modules: the CNN backbone, the Cross-Connection Layer (CCL), and the Multi-Scale Feature Aggregation (MSFA) module. These modules work together to predict both the segmentation mask and its corresponding edge during training, while only the segmentation mask is used during testing.

1. CNN Backbone :

- The input skin lesion image undergoes processing through a CNN backbone structure to generate intermediate feature maps (denoted as F) for subsequent edge and segmentation mask predictions.
- In this study, the ResNet-101 architecture with a Pyramid Pooling Module (PPM) is employed as the backbone, which is resized to 448x448 pixels for input consistency.
- Modifications are made to the Convolutional (Conv) 4 layer of ResNet-101 to maintain feature resolution, utilizing a stride of 1 and a dilate rate of 2.
- Details of the architecture, including filter sizes, number of filters, and operations within each block, are provided in Table 1. The PPM module fuses features from four different pyramid scales to enrich context information.

2. Cross-Connection Layer (CCL) :

- The method incorporates two parallel branches for edge and segmentation mask prediction. Interactions between these branches are facilitated by the CCL, where intermediate feature maps of each task are fed as inputs to the subsequent sub-block of the other task.
- This cross-connection allows for leveraging edge information during segmentation mask prediction and vice versa, promoting implicit regularization of mask boundaries and improved edge learning.

- The CCL consists of sequential convolutional blocks for both the Seg and Edge subnets, implemented using a residual block structure. Details of the residual block are provided, including scaling input feature maps, channel shrinking, and up sampling to maintain feature map size.

3. Multi-Scale Feature Aggregation (MSFA) Module :

- The MSFA module aggregates feature maps from multiple resolutions to make the final prediction. It utilizes a Conv block to generate prediction maps at each resolution.
- Feature maps from different resolutions are fed into the MSFA module, where predictions are made at each scale. These predictions are then up sampled, concatenated, and aggregated into a final prediction using convolutional operations with 1x1 kernels.
- Shared parameters are used in the MSFA module for both segmentation and edge prediction, encouraging shared quality between the predicted masks and edges.

Overall, the proposed method combines these modules to predict segmentation masks and edges simultaneously during training, enhancing model performance and robustness. The detailed architecture and integration of auxiliary tasks contribute to the effectiveness of the proposed approach for skin lesion segmentation.

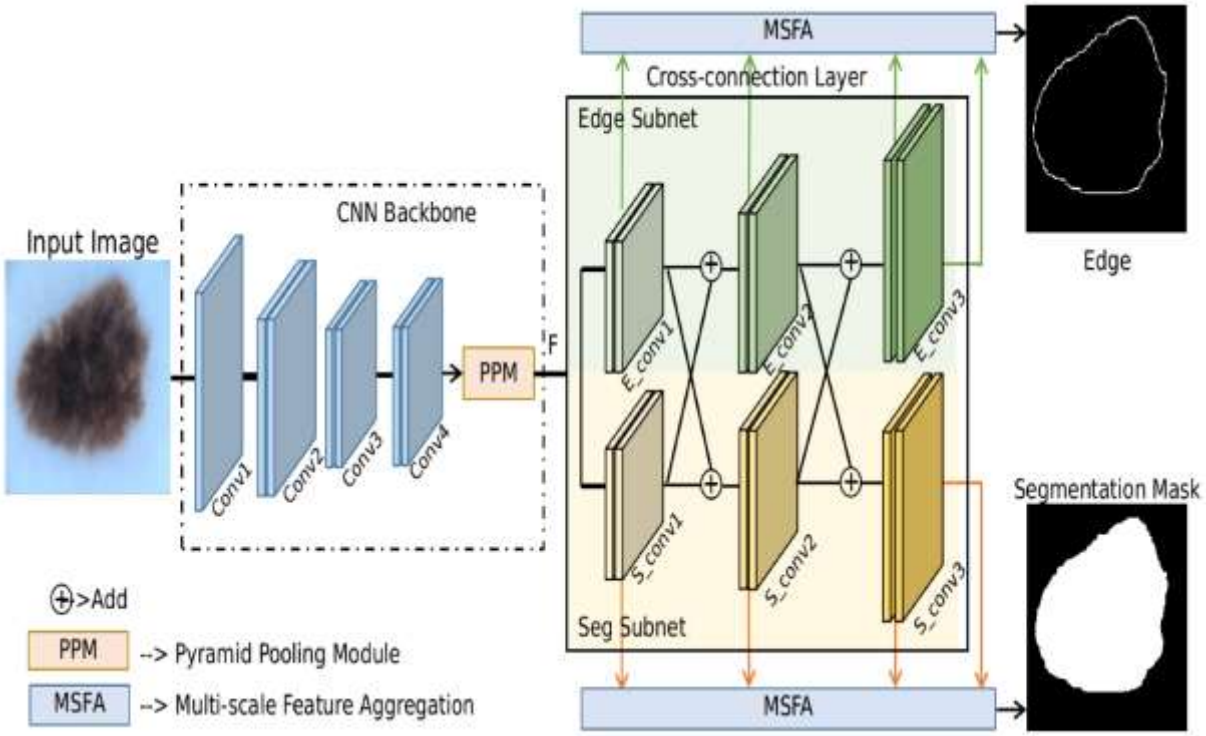

Fig* 2.2.3 *: Schematic Diagram for the Proposed Method

2.3 Fast level set method for glioma brain tumor segmentation based on Super pixel fuzzy clustering and lattice Boltzmann method

2.3.1 Introduction

Segmentation of glioma brain tumors from medical images is critical for diagnosis, treatment planning, and monitoring disease progression. However, accurately delineating gliomas poses challenges due to their irregular shapes, fuzzy boundaries, and intensity variations in imaging modalities such as MRI scans. Traditional segmentation methods often struggle with these complexities, leading to inaccurate results and potentially impacting clinical decisions.

To address these challenges, this study proposes a novel approach that integrates two advanced techniques: Super pixel Fuzzy Clustering and the Lattice Boltzmann Method (LBM). This approach aims to improve the accuracy and efficiency of glioma segmentation by leveraging the complementary strengths of these methods.

The proposed method combines super pixel-based fuzzy clustering, which enhances robustness by accommodating uncertainty in pixel memberships and preserving spatial coherence, with the LBM, a computational fluid dynamics approach adapted for image segmentation. By modeling the propagation of level set contours, LBM offers advantages in accurately delineating gliomas with irregular shapes and fuzzy boundaries, while also enabling faster segmentation compared to traditional methods.

The primary objective of this study is to develop a fast and accurate glioma segmentation method capable of handling the challenges posed by irregular tumor morphology and intensity variations in medical images. Experimental validation will be conducted using a diverse dataset of MRI scans, with the proposed method compared against state-of-the-art segmentation approaches.

2.3.2 Merits , Demerits and Challenges

Merits :

Merits of Fast Level Set Method for Glioma Brain Tumor Segmentation based on Superpixel Fuzzy Clustering and Lattice Boltzmann Method:

1. Improved Accuracy : The integration of super pixel-based fuzzy clustering and the Lattice Boltzmann Method enhances the accuracy of glioma segmentation by effectively capturing the irregular shapes and fuzzy boundaries characteristic of gliomas. This results in more precise and reliable tumor delineation, aiding clinicians in accurate diagnosis and treatment planning.

2. Efficiency : Leveraging the parallelizable nature of the Lattice Boltzmann Method, the proposed method offers faster segmentation compared to traditional techniques. This efficiency is crucial for clinical applications where rapid processing of medical images is essential for timely decision-making, enabling quicker diagnosis and treatment initiation for patients with gliomas.

3. Robustness : Super pixel fuzzy clustering accommodates uncertainty in pixel memberships, making the segmentation method more robust to intensity variations and noise in medical images. This ensures consistent and reliable segmentation results across different imaging modalities and patient datasets, contributing to the method's reliability in diverse clinical scenarios.

4. Adaptability : The modular nature of the proposed approach allows for easy adaptation to different imaging modalities and tumor types. This flexibility ensures applicability across a wide range of clinical scenarios, enhancing its utility in various healthcare settings and facilitating seamless integration into existing workflows for glioma diagnosis and treatment.

Demerits :

1. Complexity: Integrating multiple techniques such as super pixel fuzzy clustering and the Lattice Boltzmann Method may introduce complexity into the segmentation pipeline. This complexity could potentially hinder the method's adoption, requiring specialized expertise for implementation and optimization, and making it less accessible to clinicians without extensive computational background.

2. Computational Resources: While the proposed method aims to be efficient, it may still demand significant computational resources, particularly for processing large-scale medical image datasets. This reliance on computational power could limit its applicability in resource-constrained environments or settings lacking access to high-performance computing infrastructure.

3. Parameter Tuning: Like many advanced segmentation methods, the proposed approach may require careful tuning of parameters to achieve optimal segmentation performance. Identifying the appropriate parameter values for different datasets and clinical scenarios may necessitate iterative experimentation and validation, potentially adding to the computational burden and time required for implementation.

4. Validation and Clinical Adoption: Validating the proposed method rigorously across diverse datasets and clinical scenarios is essential to ensure its reliability and generalizability. Clinical adoption of the method may also require regulatory approval and integration into existing healthcare workflows, which could present logistical and administrative challenges, slowing down its adoption in clinical practice.

Challenges:

Model Complexity and Overfitting: Despite efforts to address overfitting through preprocessing and dropout layers, deep learning models with attention mechanisms can still be prone to overfitting, especially when trained on limited data. Balancing model complexity with generalization capability

2.3.3 Implementation

Implementing the Fast Level Set Method for Glioma Brain Tumor Segmentation based on Super pixel Fuzzy Clustering and Lattice Boltzmann Method involves several key steps:

Firstly, preprocess the medical imaging data (e.g., MRI scans) to enhance contrast, reduce noise, and standardize intensity levels. This may involve skull stripping, bias correction, intensity normalization, and image registration. Next, apply a super pixel segmentation algorithm to divide the preprocessed images into homogeneous regions or super pixels. Implement fuzzy clustering to assign each super pixel to multiple tumor classes based on intensity and spatial features, accounting for uncertainty in pixel memberships.

Adapt the Lattice Boltzmann Method for image segmentation by formulating the level set evolution equation within the LBM framework. Define the lattice structure and collision rules to simulate the propagation of level set contours, leveraging parallelization for computational efficiency. Initialize the level set function representing the evolving tumor boundary and set parameters for the level set evolution, such as regularization term, speed function, and stopping criteria.

Implement the numerical scheme for evolving the level set function over iterations, updating it based on the defined speed function to guide contour evolution towards tumor boundaries. Incorporate super pixel-based fuzzy clustering and Lattice Boltzmann Method framework into this process for accurate delineation of glioma boundaries.

Post-process the segmented tumor regions using morphological operations to smooth contours, remove small artifacts, and improve segmentation accuracy. Optionally, perform additional post-processing steps such as region growing or boundary refinement
Validate the implemented method using ground truth annotations or expert evaluations to assess segmentation accuracy and performance metrics. Evaluate computational efficiency by comparing processing times against existing segmentation techniques.

Integrate the implemented method into existing medical image analysis workflows or develop a standalone software tool for clinical use. Ensure compatibility with standard image formats and provide user-friendly interfaces for data input, parameter adjustment, and result visualization. Throughout the implementation process, maintain documentation and code organization to facilitate reproducibility, collaboration, and future enhancements. Continuous testing and validation against diverse datasets are essential to ensure method robustness and reliability across different clinical scenarios.

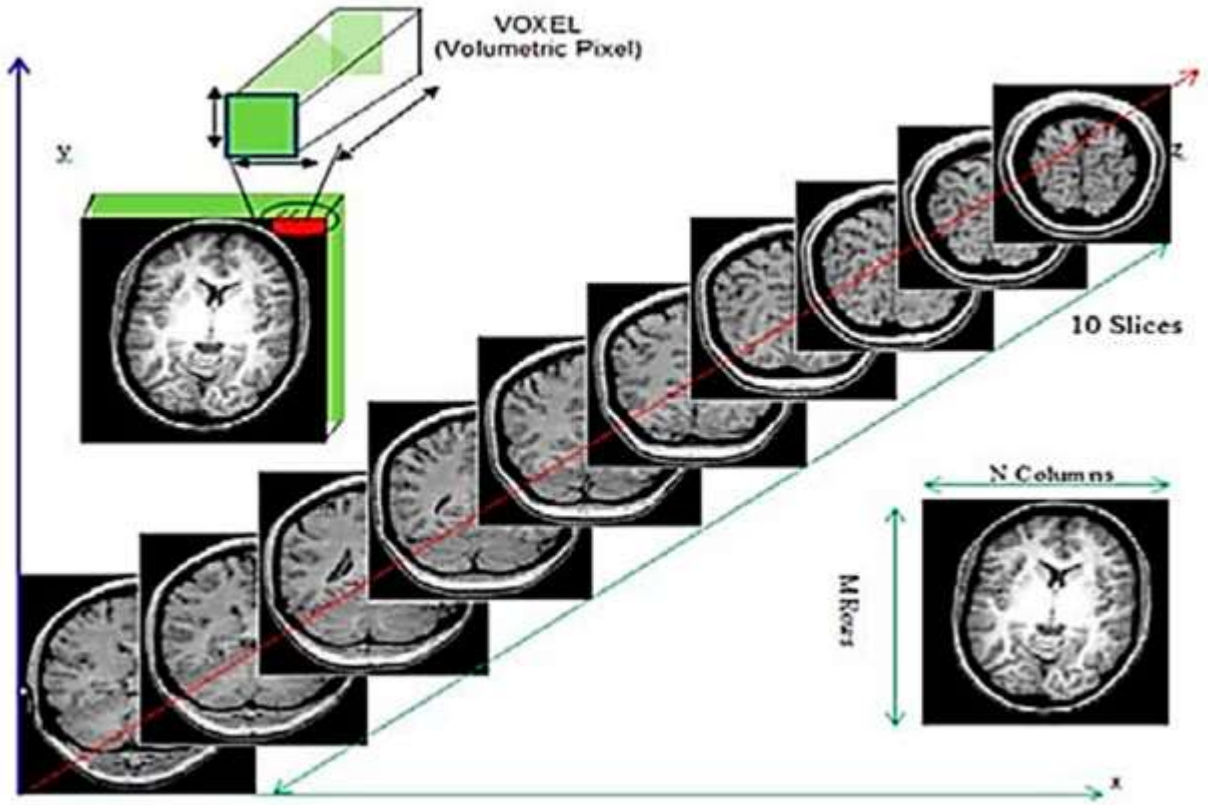

Fig* 2.3.3 : *Voxel and Slice in MRI Data

CHAPTER 3
PROPOSED SYSTEM

CHAPTER 3

PROPOSED SYSTEM

3.1 Objective of the Proposed System

The objective of the project is to develop a biomedical image segmentation and detection system using U-Net architecture for identifying and delineating brain tumors and skin lesions. This project aims to leverage deep learning techniques to assist medical professionals in accurately locating and analyzing abnormalities in medical images, thereby aiding in diagnosis and treatment planning. The project aims to develop a sophisticated biomedical image analysis system that empowers medical professionals with advanced tools for early detection, diagnosis, and treatment monitoring of brain tumors and skin lesions, ultimately improving patient outcomes and healthcare delivery.

3.2 Algorithms Used for Proposed Model

3.2.1 U-Net Algorithm

UNET is a convolutional neural network architecture that was developed by Bio-Medical Image Segmentation by Olaf Ronneberger in the year 2015 at the university Freiburg in Germany. It is most populary used for any image segmentation tasks. It is Fully Convolutional neural networks that is designed to learn for fewer training samples.
U-Net gets its name from its architecture. The “U” shaped model comprises convolutional layers and two networks. First is the encoder, which is followed by the decoder.

U-Net stands out as a specialized architecture tailored specifically for semantic segmentation tasks. Its name originates from its unique architectural design resembling a "U." This model comprises convolutional layers intricately connected in two networks. Its architecture features a contracting path, dedicated to capturing contextual information, and an expansive path, focused on refining spatial details. This distinctive layout enables the efficient integration of both local and global features from the input data, facilitating robust segmentation.

We've crafted our implementation of the U-Net architecture utilizing Python alongside prevalent deep learning libraries. Within our system, we input RGB MRI images depicting brain tumours or RGB images depicting skin lesions, each containing 3 channels. Our aim isn't to classify these images but rather to produce a mask identical in size to the input image.

To achieve our goal, we utilize an encoder network, also referred to as the contracting network. Its main purpose is to reduce the spatial dimensions of the input while increasing the number of channels. This network comprises four encoder blocks, each composed of two convolutional layers with a 3x3 kernel size and valid padding, followed by a Rectified Linear Unit (ReLU) activation function. Afterward, the output from each block undergoes a max-pooling layer with a 2x2 kernel size and a stride of 2. This max-pooling operation effectively decreases the spatial dimensions, aiding in reducing computational costs during training. The final output of the encoder layer is an image with dimensions 16x16x1024, representing a feature map of the input image. Convolutional operations within the encoder are depicted by blue arrows, while max-pooling operations are indicated by red arrows in the model architecture. This architectural layout facilitates the extraction of hierarchical features from the input images, preparing them for subsequent processing in the decoder section of the U-Net architecture.

Situated between the encoder and decoder networks, the bottleneck layer occupies a central position within the architecture. Comprising two convolutional layers followed by Rectified Linear Unit (ReLU) activation, this layer serves as the final representation of the feature map.

The decoder network, also termed the expansive network, operates in opposition to the encoder, expanding spatial dimensions while reducing channel numbers. This entails increasing the size of the feature maps to match the input image's dimensions, as depicted by green arrows in the architectural diagram. By leveraging skip connections, the decoder network receives the feature map from the bottleneck layer and generates a segmentation mask. It consists of four decoder blocks, initiating with a transpose convolution using a 2x2 kernel size and stride of 2. The resultant output is fused with the corresponding skip layer connection from the encoder block. Subsequently, two convolutional layers employing a 3x3 kernel size are applied, followed by a ReLU activation function, ultimately producing an image with dimensions 256x256x64.

Skip connections, illustrated as black arrows in the architectural diagram, play a crucial role in leveraging contextual feature information from encoder blocks to generate the segmentation map. These connections combine high-resolution features from the encoder with the feature map output of the bottleneck layer in the decoder. This fusion aids in restoring spatial details that may have been lost during down sampling in the encoder, enabling image reconstruction or segmentation at the original resolution. By utilizing this amalgamated information, the decoder conducts up sampling to produce a high-resolution output resembling the initial input. Furthermore, following the last decoder block, a 1x1 convolution, indicated by violet arrows in the model's structure, is employed alongside a sigmoid activation. This generates a segmentation mask output sized 256x256x1, containing pixel-wise classifications. This mechanism enables the transmission of information from the contracting path to the expansive path, capturing both feature details and localization. Consequently, it enhances the effectiveness of U-Net for semantic segmentation tasks.

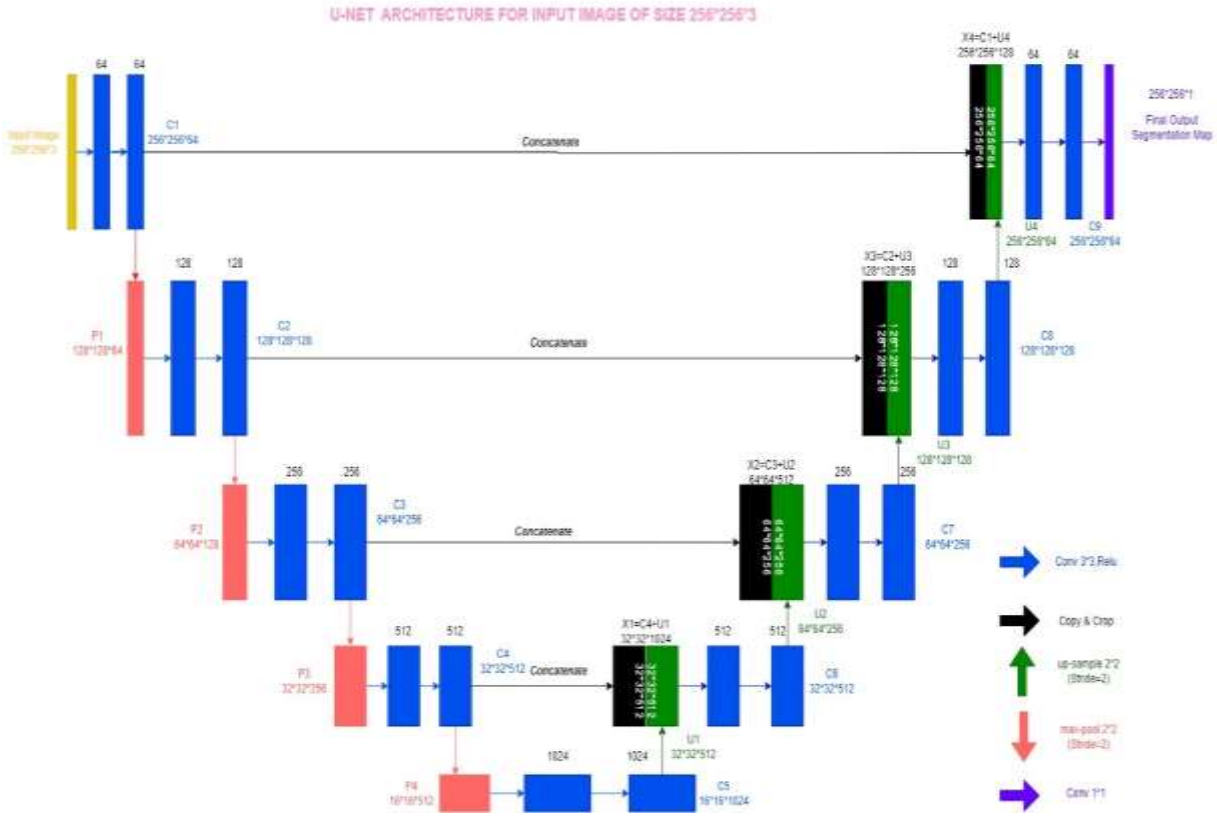

Fig 3.2.1 : Proposed U-Net Architecture

3.3 Designing

The system's user interface is tailored to cater to two distinct medical conditions: brain tumors and skin lesions. Users engage with the system by uploading medical images pertinent to these conditions. The system's output comprises masked images that pinpoint and isolate the identified areas of interest, streamlining the process for in-depth analysis.

The foundation of this system for biomedical image segmentation relies on the U-Net model, a convolutional neural network architecture recognized for its excellence in semantic segmentation tasks. U-Net's standout feature lies in its remarkable precision in delineating complex structures within medical images, making it a preferred choice for tasks such as identifying brain tumors and skin lesions. The system utilizes Django framework for web development, employing HTML, CSS, and JavaScript for front-end functionalities, while relying on an MSSQL database for efficient data management. Python powers the back-end operations, facilitating the processing of images. This configuration seamlessly integrates the U-Net model into a user-friendly web application designed for medical image segmentation.

In this project, the front end makes use of HTML, CSS, and JavaScript, while Python is employed for the back end. The system's functionality is backed by the Django framework and a MySQL Database, providing a robust infrastructure

3.3.1 UML Diagrams

Use case Diagram :

A use case diagram outlines the interactions between actors (users or systems) and the system under consideration. In the case of biomedical image segmentation for brain tumors and skin lesions using a U-Net model, the actors and use cases can be identified. This use case diagram illustrates how U-Net-based image segmentation is utilized in the biomedical domain for detecting brain tumors and skin lesions, facilitating diagnosis and treatment planning by healthcare professionals. This use case diagram illustrates the interactions between users and the system for biomedical image segmentation using the U-Net model for brain tumors and skin lesions.

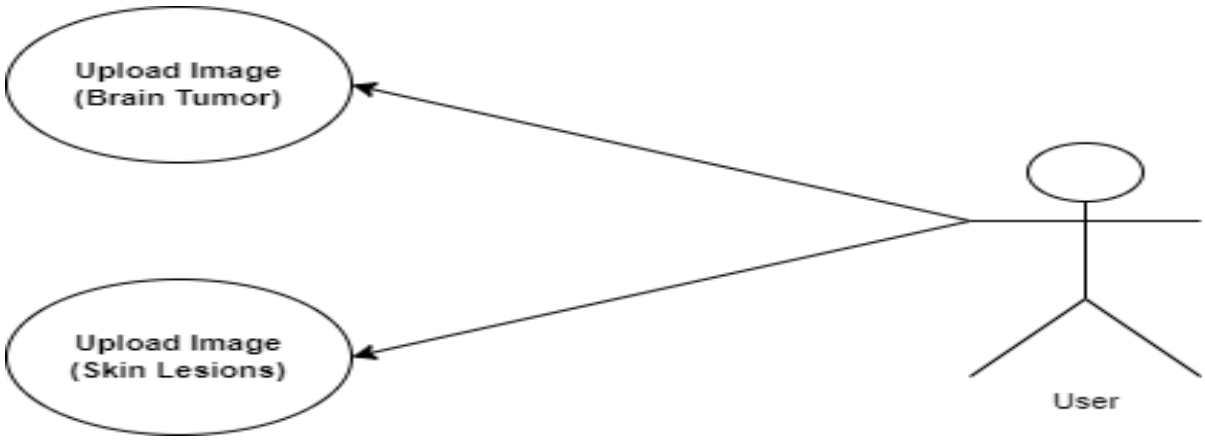

Fig* 3.3.1.1 *: Use Case Diagram

Sequence Diagram :

This sequence diagram outlines the key steps involved in biomedical image segmentation using U-Net for brain tumor and skin lesion diseases. It demonstrates the flow of data and operations within the segmentation pipeline, from image acquisition to output visualization. Below is a simplified sequence diagram illustrating the process of biomedical image segmentation using U-Net for brain tumor and skin lesion diseases.

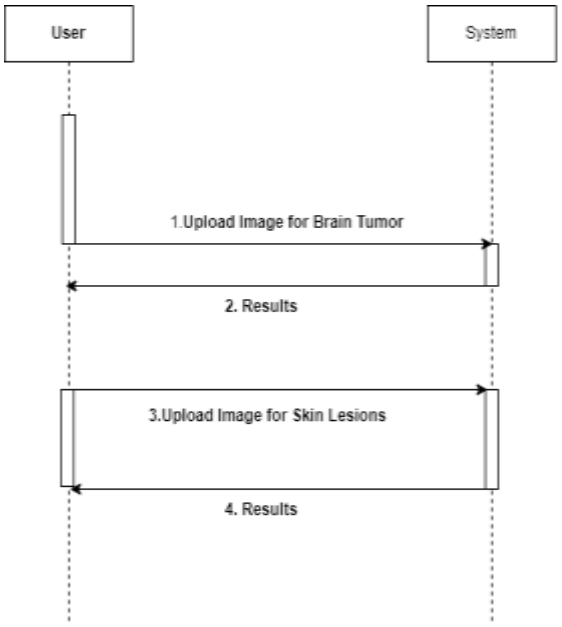

Fig* 3.3.1.2 *: Sequence Diagram

Activity Diagram :

Creating an activity diagram for user interaction in biomedical image segmentation, particularly for brain tumor and skin lesion diseases using a U-Net architecture, involves illustrating the steps and interactions involved in the segmentation process. Below is an ctivity diagram depicting the user's interaction.

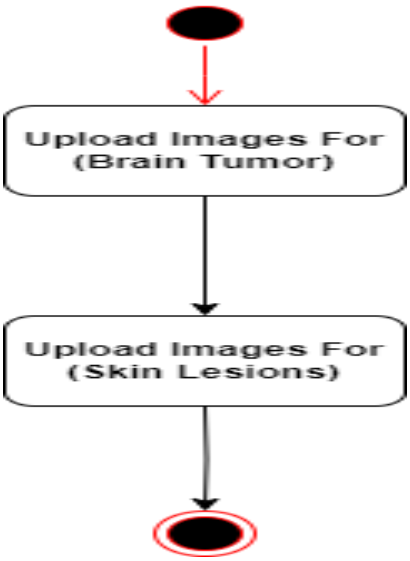

Fig* 3.3.1.3 *:Activity Diagram

Data Flow Diagram (DFD's) :

A Data Flow Diagram (DFD) is a graphical representation of the flow of data through a system. In the context of biomedical image segmentation for brain tumor and skin lesion diseases using UNet, we can create a DFD to illustrate how data flows through the various components of the system. Here's a simplified DFD focusing on the user's interaction with the system.

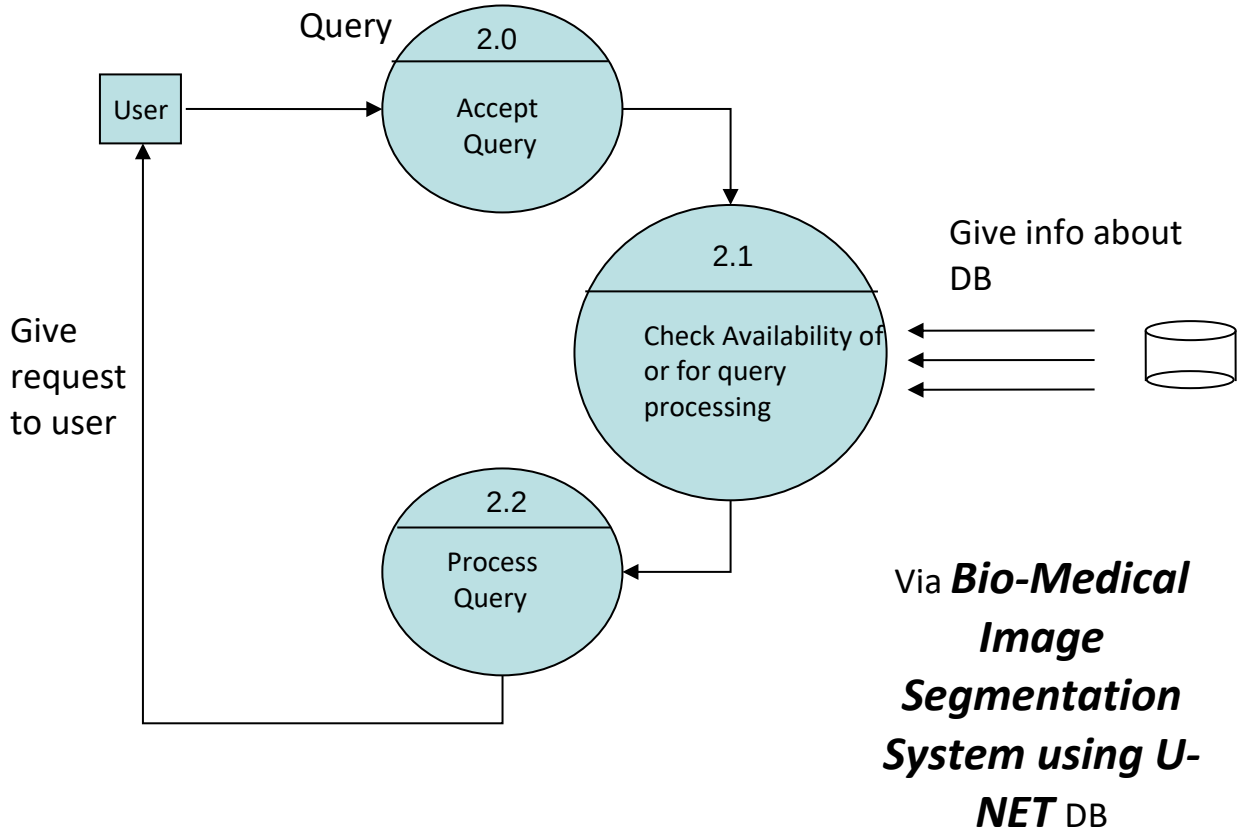

Fig* 3.3.1.4 *: Data Flow Diagram

3.4 Stepwise Implementation and Code

3.4.1 Implementation Steps :

Here's a step-by-step guide on implementing a web-based application for biomedical image segmentation using U-Net for brain tumor and skin lesion diseases. This application will have a frontend implemented using HTML, CSS, and JavaScript, a backend using Django, and the segmentation model implemented in Python. Additionally, we used XAMPP as the server environment.

Step 1: Set Up the Environment

- Install Python and Django.
- Install necessary Python libraries such as TensorFlow, Keras, Django, etc.
- Install XAMPP for setting up the server environment.

Step 2 : Implement the U-Net Model

- Write Python code to implement the U-Net architecture for biomedical image segmentation.
- Train the model using appropriate datasets for brain tumor and skin lesion segmentation.
- Save the trained model weights for later use.

Step 3: Implement the Django Backend

- Set up a Django project.
- Create Django views to handle incoming requests from the frontend.
- Implement necessary logic to preprocess images, perform segmentation using the trained U-Net model, and return the segmented images.
- Integrate the code with the trained U-Net model to perform segmentation.

Step 4: Implement the Frontend

- Design the user interface using HTML for structure, CSS for styling, and JavaScript for interactive elements.
- Create an HTML form for users to upload images.
- Use JavaScript to send AJAX requests to the Django backend with the uploaded images.
- Display the segmented images returned by the backend on the frontend interface.

Step 5: Integrate Frontend with Django Backend

- Configure Django to serve static files (HTML, CSS, JS).
- Set up Django URL patterns to route requests from the frontend to the appropriate views.
- Ensure that the Django views return appropriate responses to the frontend.

Step 6: Test the Application

- Test the application by uploading sample biomedical images for segmentation.
- Verify that the segmentation results are accurate and displayed correctly on the frontend.

Step 7: Deployment

- Set up XAMPP to host the Django application.
- Configure the server to handle incoming requests.
- Deploy the application to the server.

Following these steps should help you create a web-based application for biomedical image segmentation using U-Net with a frontend implemented in HTML, CSS, and JavaScript, a backend in Django, and hosted using XAMPP server and also U-Net Implementation in python for performing accurate biomedical image segmentation and detection.

3.4.2 Datasets Used in the Project

Our application focuses on predicting brain tumour locations using publicly accessible MRI datasets from The Cancer Imaging Archive (TCIA). These datasets contain manually annotated FLAIR abnormality segmentation masks from 110 patients in The Cancer Genome Atlas (TCGA) lower-grade glioma collection. Each image, sized at 256x256 pixels, merges MRI slices from three modalities into RGB format. Furthermore, we leverage an additional dataset for skin lesion classification, combining data from HAM10000 (2019) and MSLDv2.0, encompassing 14 different types of skin lesions.

3.4.3 Steps To Run The Application :

1. Open XAMPP and start both Apache and MySQL services.
2. Navigate to the directory "C:\WORKSPACE\Brain_Skin_Mask" using Command Prompt.
3. Run the following command to start the application:
"python manage.py runserver"
4. Copy the URL displayed in the Command Prompt after running the server.
5. Paste the copied URL into your web browser's address bar.
6. Once the application loads, test it by providing input images of MRI scans for brain tumor detection and skin images for skin lesion disease detection.

3.4.4 Sample Project Code :

U-Net_Model.Py :

```
import torch
import torch.nn as nn
import torch.nn.functional as F
import torch.optim as optim

class UNet(nn.Module):
    def __init__(self, n_channels =3, n_classes=1):
        # Initialize the nn.Module base class
        super(UNet, self).__init__()
        self.n_channels = n_channels
        self.n_classes = n_classes

        # Contracting path (encoder)
        self.conv1 = nn.Conv2d(self.n_channels, 64, kernel_size=3, padding=1)
        self.conv2 = nn.Conv2d(64, 128, kernel_size=3, padding=1)
        self.conv3 = nn.Conv2d(128, 256, kernel_size=3, padding=1)
        self.conv4 = nn.Conv2d(256, 512, kernel_size=3, padding=1)
        self.conv5 = nn.Conv2d(512, 1024, kernel_size=3, padding=1)
        self.pool = nn.MaxPool2d(kernel_size=2, stride=2)

        # Expansive path (decoder)
        self.upconv1 = nn.ConvTranspose2d(1024, 512, kernel_size=2, stride=2)
        self.conv6 = nn.Conv2d(1024, 512, kernel_size=3, padding=1)
        self.upconv2 = nn.ConvTranspose2d(512, 256, kernel_size=2, stride=2)
        self.conv7 = nn.Conv2d(512, 256, kernel_size=3, padding=1)
        self.upconv3 = nn.ConvTranspose2d(256, 128, kernel_size=2, stride=2)
```

```
self.conv8 = nn.Conv2d(256, 128, kernel_size=3, padding=1)
        self.upconv4 = nn.ConvTranspose2d(128, 64, kernel_size=2, stride=2)
        self.conv9 = nn.Conv2d(128, 64, kernel_size=3, padding=1)
        self.conv10 = nn.Conv2d(64, self.n_classes, kernel_size=1)

        self.optimizer = optim.Adam(self.parameters(), lr=0.0003)
        self.criterion = nn.BCEWithLogitsLoss()
        self.scaler = torch.cuda.amp.GradScaler()

    def forward(self, x):
        # Contracting path (encoder)
        x1 = F.relu(self.conv1(x))
        x2 = F.relu(self.conv2(self.pool(x1)))
        x3 = F.relu(self.conv3(self.pool(x2)))
        x4 = F.relu(self.conv4(self.pool(x3)))
        x5 = F.relu(self.conv5(self.pool(x4)))
        # Expansive path (decoder)
        x6 = F.relu(self.upconv1(x5))
        x6 = torch.cat([x4, x6], dim=1)
        x6 = F.relu(self.conv6(x6))
        x7 = F.relu(self.upconv2(x6))
        x7 = torch.cat([x3, x7], dim=1)
        x7 = F.relu(self.conv7(x7))
        x8 = F.relu(self.upconv3(x7))
        x8 = torch.cat([x2, x8], dim=1)
        x8 = F.relu(self.conv8(x8))
        x9 = F.relu(self.upconv4(x8))
        x9 = torch.cat([x1, x9], dim=1)
        x9 = F.relu(self.conv9(x9))
        x10 = self.conv10(x9)
        return x10
```

View.Py :

```
from django.shortcuts import render,redirect
import numpy as np
from django.http import HttpResponse
from PIL import Image
import numpy as np
import torch
import matplotlib.pyplot as plt
from home.Unet_model import UNet
import torch
import torch.nn as nn
import torch.nn.functional as F
import torch.optim as optim
import datetime

DEVICE =  torch.device("cpu")
#brain_model = torch.load("home/brain_tumour_entire_model.pth")
brain_model = torch.load("home/brain_tumour_entire_model.pth",
map_location=torch.device('cpu'))
brain_model.to("cpu")
brain_model.eval()

skin_model = UNet().to(DEVICE)
#skin_model.load_state_dict(torch.load('home/checkpointN20_.pth (1).tar')['state_dict'])
skin_model.load_state_dict(torch.load('home/checkpointN20_.pth (1).tar',
map_location=torch.device('cpu'))['state_dict'])
skin_model.eval()

def index(request):
    return render(request,'index.html')
```

```
def brain_tumour(request):
    if request.method == "POST":
        brain_image = request.FILES['brain_image']
        brain_input = preprocess_image(brain_image)
        x_tensor = torch.from_numpy(brain_input)
        x= x_tensor.to('cpu')
        x = x.float()
        prediction = brain_model.predict(x)
        prediction = torch.where(prediction > 0.5, 1, 0)
        prediction1 = prediction.to('cpu')[0][0]
        mask_array = prediction1.cpu().numpy()
        result = "NO TUMOUR DETECTED"
        mask_rgb = np.stack((mask_array, mask_array, mask_array), axis=-1)
        overlay_image = image_show(brain_image)  # Create a copy of the original image
        print(overlay_image.shape)
        print(mask_rgb.shape)
        for y in range(overlay_image.shape[0]):
            for x in range(overlay_image.shape[1]):
                if mask_array[y, x] == 1:
                    overlay_image[y, x, :] = [1, 0, 0]
                    result = "TUMOUR DETECTED"

        print(overlay_image.shape)
        fig, ax = plt.subplots(1, 3, figsize=(15, 5))
        fig.suptitle(result, fontsize=16)
        ax[0].imshow(image_show(brain_image))
        ax[0].set_title('Original Image')
            ax[1].imshow(mask_array)
        ax[1].set_title('Generated Mask')
        ax[2].imshow(overlay_image)
```

```
        ax[2].set_title('Detected Tumour')
        now = datetime.datetime.now()
        unique_name = f"plot_brain_{now.strftime('%Y%m%d_%H%M%S')}.png"
        plt.savefig(f"media/{unique_name}")
        #plt.show()
        plot_path = f"/media/{unique_name}"
        return render(request, 'result.html', {'plot_path': plot_path})
    return render(request,'brain_tumour.html')

def image_show(image_path, target_size=(256, 256)):
    image = Image.open(image_path)
    image = image.resize(target_size)
    image_array = np.array(image) / 255.0
    return image_array
def preprocess_image(image_path, target_size=(256, 256)):
    image = Image.open(image_path)
    image = image.resize(target_size)
    image_array = np.array(image) / 255.0
    if len(image_array.shape) == 2:
        image_array = np.stack((image_array,) * 3, axis=-1)
    image_array = np.expand_dims(image_array.transpose(2, 0, 1), axis=0)
    return image_array

def skin_lease(request):
    return render(request,'index.html')
def skin_lease(request):
    if request.method == "POST":
        skin_image = request.FILES['skin_image']
```

```
    resized_image = preprocess_image(skin_image)
    img_tensor = torch.Tensor(resized_image).to(DEVICE)
    #predicted_array = getpredictionskin.getpred(img_tensor)
    generated_mask = skin_model(img_tensor).squeeze().cpu()
    predicted_array = (generated_mask > 0.5).float().detach().numpy()
    overlay_image = image_show(skin_image)
    result = "NO SKIN LESIONS DETECTED"
    for y in range(overlay_image.shape[0]):
        for x in range(overlay_image.shape[1]):
            if predicted_array[y, x] == 1:
                overlay_image[y, x, :] = [1, 0, 0]
                result = "SKIN LESIONS DETECTED"

    print(overlay_image.shape)

    fig, ax = plt.subplots(1, 3, figsize=(15, 5))
    fig.suptitle(result, fontsize=16)
    ax[0].imshow(image_show(skin_image))
    ax[0].set_title('Original Image')
    ax[1].imshow(predicted_array)
    ax[1].set_title('Generated Mask')
    ax[2].imshow(overlay_image)
    ax[2].set_title('Detected Lesion')
    now = datetime.datetime.now()
    unique_name = f"plot_skin_{now.strftime('%Y%m%d_%H%M%S')}.png"
    plt.savefig(f"media/{unique_name}")

     #plt.show()
    plot_path = f"/media/{unique_name}"
    return render(request, 'result.html', {'plot_path': plot_path})

  return render(request,'skin_lease.html')
```

CHAPTER 4
RESULTS AND DISCUSSION

CHAPTER 4

RESULTS AND DISCUSSION

The outcome of our proposed implementation involves selecting between brain tumors or skin lesions. Afterward, we upload the image or input images. From the input image, we receive results indicating the detected tumor or lesion region, along with a masked image. Additionally, we receive information regarding the presence of tumors or lesions in the input image. If no tumor or lesion is detected, we obtain an empty masked image without any segmented regions of tumors or lesions. In such cases, no detected tumor or lesion regions are displayed in the image.

4.1 Output Screens

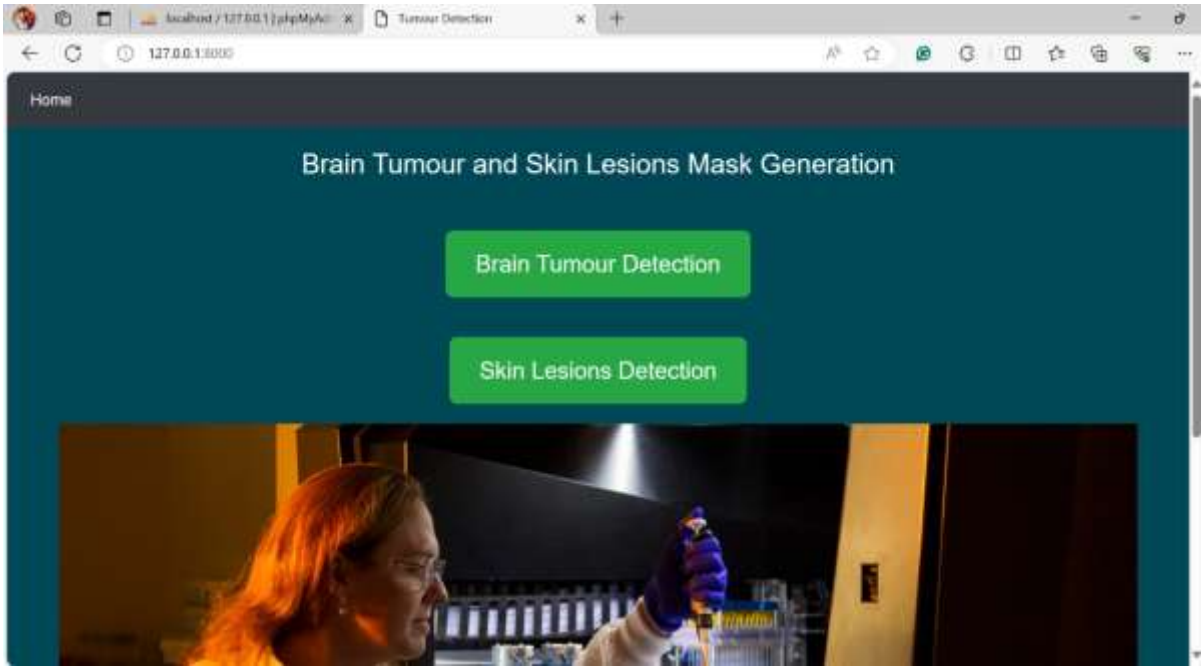

Fig* 4.1.1 *: Bio-Medical Image segmentation and Detection website

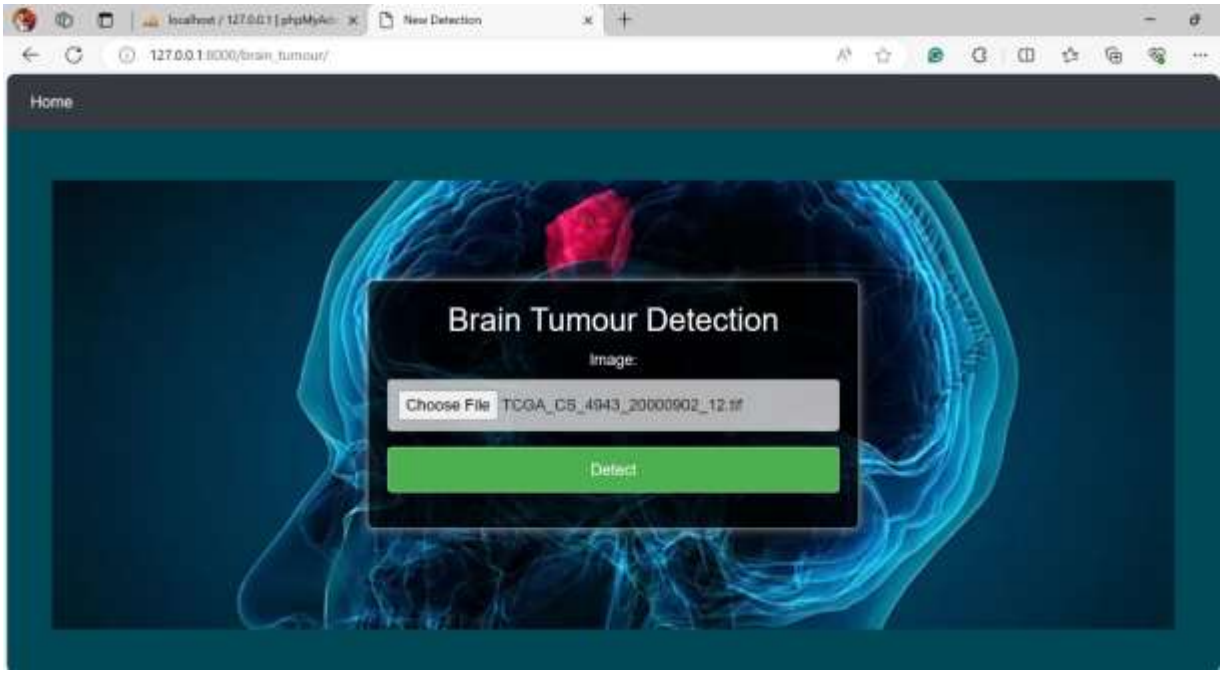

Fig 4.1.2 : Brain Tumour Disease Detection

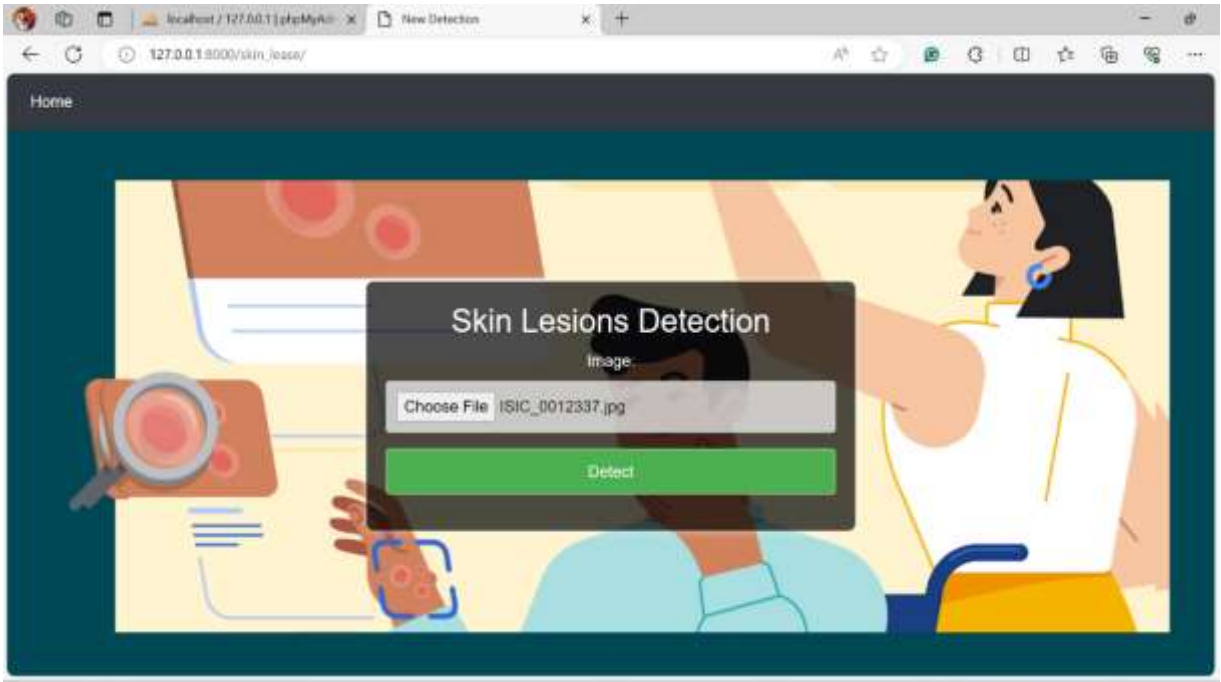

Fig 4.1.3 : Skin Lesion Disease Detection

The illustration depicts the identified location of a brain tumor within the input image and its corresponding representation are the segmented region highlighting the tumor area extracted from the input image.

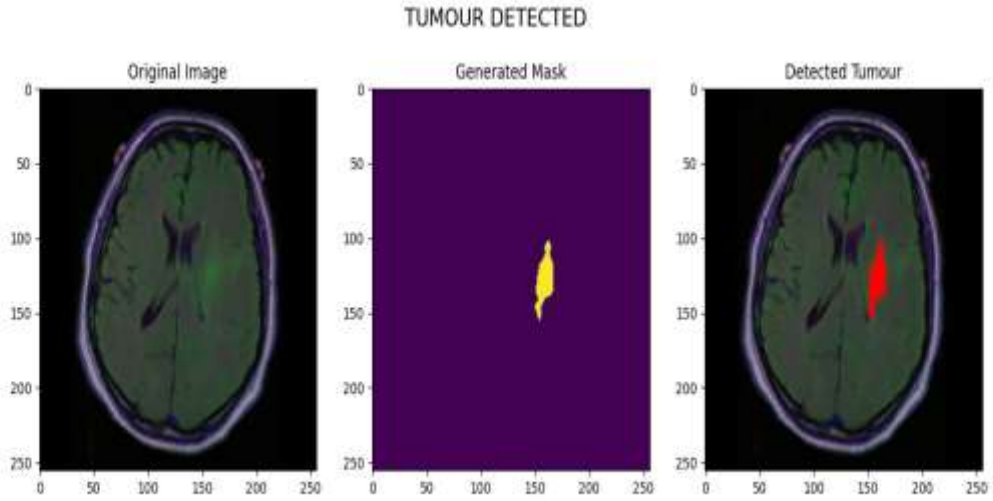

Fig* 4.1.4 *: Brain Tumour Detection

The figure showcases the identified the position of a skin anomaly within the input image, along with the delineated area extracted from the input image.

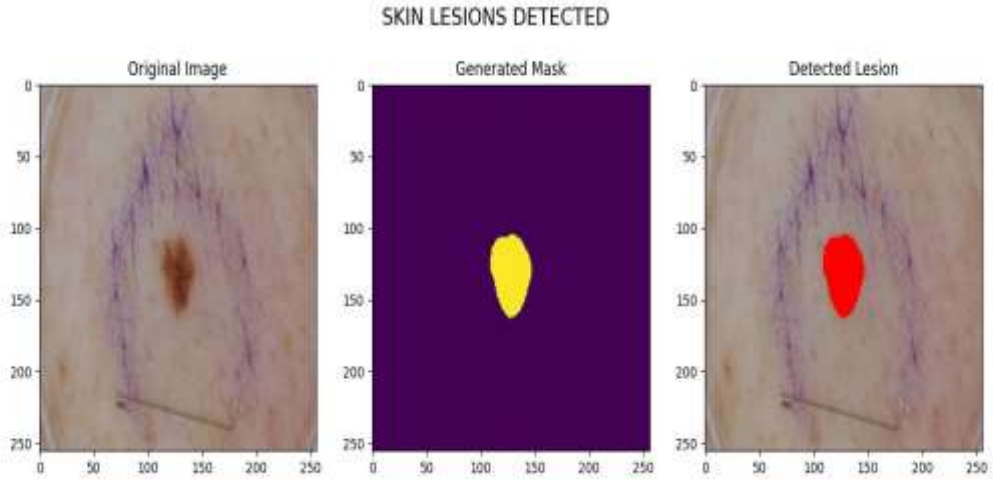

Fig* 4.1.5 *: Skin Lesion Detection

4.2 Performance Metrics :

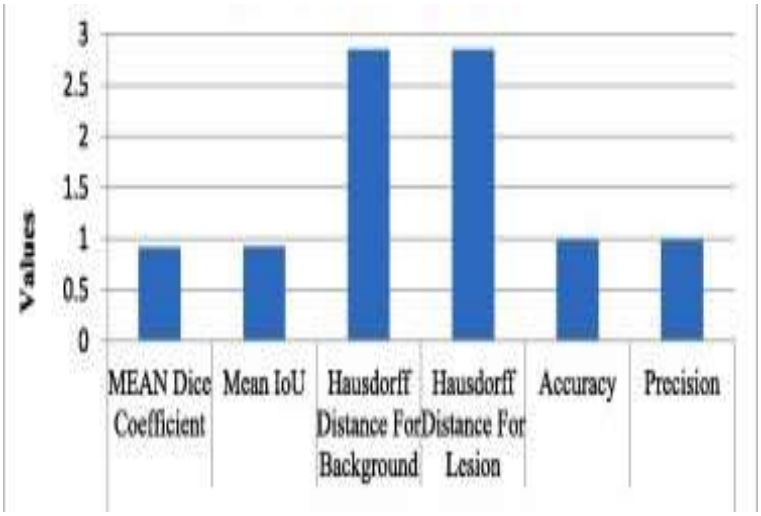

Fig 4.2.1 Performance metrics

CHAPTER 5
CONCLUSION

CHAPTER 5
CONCLUSION

Our implementation of the U-Net architecture prioritizes lightweight design without compromising segmentation accuracy, thereby minimizing the need for extensive data augmentation. This framework shows potential for integration into medical settings, where trained physicians could utilize it as a supplementary tool for evaluating patients' MRI or skin images. While investigations into brain tumour segmentation using deep learning have made considerable strides, additional research is necessary to further advance the field warranted to enhance the network's performance, particularly in reducing false negatives and false positives in biomedical image analysis. Future research endeavors could explore the incorporation of disease-specific information for brain tumours or skin lesions, potentially offering substantial benefits to the medical community

REFERENCES

REFERENCES

[1] Ramin Ranjbarzadeh et al,"Brain tumor segmentation built upon DL techniques and an attention mechanism using MRI multi-modalities brain images", Scientific Reports,11 Article number 10930 , May 2021.

[2] Lina Liu Ying Y.Tsui et al,"Skin Segmentation of Lesions Utilizing DL with Auxiliary Task",J Imaging,April 2021.

[3] Esteva A., Kuprel B., Novoa R.A., Ko J., Swetter S.M., Blau H.M.," Thrun Sophisticated classification of skin cancer akin to that of dermatologists achieved through deep neural networks. Nature". 2017;542:115–118. Aug 2023,

[4] Khosravanian, A., Rahmanimanesh, M., Keshavarzi, P. & Mozaffari, "Fast level set method for Segmentation of glioma brain tumors utilizing a specific approach superpixel fuzzy clustering and lattice boltzmann method". Comput. Methods Programs Biomed. 198, 105809,2020.

[5] Hihao ,"Medical Image Segmentation Derived from U-Net", Journal of Physics:Conference Series,vol.2547 , 2023

[6] Mohammed Khouy ,"Medical Image Segmentation Using Automatic Optimized U-Net Architecture Based on Genetic Algorithm",MDPI,Aug 2023.

[7] cafer budak,"Biomedical Image Partitioning Using Modified U-Net",IIETA,March 2023.

[8] Olaf Ronneberger,"U-Net: Convolutional Networks for Biomedical Imaging Segmentation",vol.9351,nov 2015.

GITHUB LINK :

https://github.com/sureshkumargummadi/Batch-62_Major-Project

DOI LINK :

https://doi.org/10.22214/ijraset.2024.59064