"Medical Imaging using Convolutional Neural Networks"

A Major Project Report Submitted to

Rajiv Gandhi Proudyogiki Vishwavidyalaya

Towards Partial Fulfillment for the Award of

Bachelor of Engineering in Computer Science Engineering

Submitted by:

Aanya Chourasiya(0827CS201003)
Abdul Rehman(0827CS201008)
Amisha Prajapati(0827CS201030)
Atharva Puranik(0827CS201049)

Guided by:

Prof. Ritika Bhatt
Computer Science and Engineering

Acropolis Institute of Technology & Research, Indore

Jul - Dec 2023

EXAMINER APPROVAL

The Major Project entitled **"Medical Imaging using Convolutional Neural Networks"** submitted by **Aanya Chourasiya (0827CS201003), Abdul Rehman (0827CS201008), Amisha Prajapati (0827CS201030), Atharva Puranik (0827CS201049)** has been examined and is hereby approved towards partial fulfillment for the award of Bachelor of Technology degree in Computer Science Engineering discipline, for which it has been submitted. It understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed, or conclusion drawn therein, but approve the project only for the purpose for which it has been submitted.

(Internal Examiner) **(External Examiner)**

Date: **Date:**

RECOMMENDATION

This is to certify that the work embodied in this major project entitled **“Medical Imaging using Convolutional Neural Networks”** submitted by **Aanya Chourasiya (0827CS201003), Abdul Rehman (0827CS201008), Amisha Prajapati (0827CS201030), Atharva Puranik (0827CS201049)** is a satisfactory account of the bonafide work done under the supervision of **Prof. Ritika Bhatt**, is recommended towards partial fulfillment for the award of the Bachelor of Technology (Computer Science Engineering) degree by Rajiv Gandhi Proudyogiki Vishwavidhyalaya, Bhopal.

(Project Guide)

(Project Coordinator)

(Dean Academics)

STUDENTS UNDERTAKING

This is to certify that the major project entitled **“Medical Imaging using Convolutional Neural Networks”** has been developed by us under the supervision of **Prof. Ritika Bhatt**. The whole responsibility of the work done in this project is ours. The sole intention of this work is only for practical learning and research.

We further declare that to the best of our knowledge; this report does not contain any part of any work which has been submitted for the award of any degree either in this University or in any other University / Deemed University without proper citation and if the same work is found then we are liable for explanation to this.

Aanya Chourasiya (0827CS201003)

Abdul Rehman (0827CS201008)

Amisha Prajapati (0827CS201030)

Atharva Puranik (0827CS201049)

Acknowledgement

We thank the almighty Lord for giving us the strength and courage to sail out through the tough and reach on shore safely.

There are a number of people without whom this project would not have been feasible. Their high academic standards and personal integrity provided us with continuous guidance and support.

We owe a debt of sincere gratitude, deep sense of reverence and respect to our guide and mentor **Prof. Ritika Bhatt,** Assistant Professor, AITR, Indore for her motivation, sagacious guidance, constant encouragement, vigilant supervision, and valuable critical appreciation throughout this project work, which helped us to successfully complete the project on time.

We express profound gratitude and heartfelt thanks to **Dr. Kamal Kumar Sethi**, Professor & Head CSE, AITR Indore for his support, suggestion, and inspiration for carrying out this project. We are very much thankful to other faculty and staff members of the department for providing us all support, help and advice during the project. We would be failing in our duty if we do not acknowledge the support and guidance received from **Dr. S.C. Sharma**, Director, AITR, Indore whenever needed. We take the opportunity to convey my regards to the management of Acropolis Institute, Indore for extending academic and administrative support and providing me all necessary facilities for the project to achieve our objectives.

We are grateful to our parents and family members who have always loved and supported us unconditionally. To all of them, we want to say "Thank you", for being the best family that one could ever have and without whom none of this would have been possible.

Aanya Chourasiya (0827CS201003), Abdul Rehman (0827CS201008),

Amisha Prajapati (0827CS201030), Atharva Puranik (0827CS201049)

Executive Summary

Medical Imaging using Convolutional Neural Networks

This project is submitted to Rajiv Gandhi Proudyogiki Vishwavidhyalaya, Bhopal (MP), India for partial fulfillment of Bachelor of Engineering in Information Technology branch under the sagacious guidance and vigilant supervision of Prof. Ritika Bhatt.

This project focuses on the application of CNN in the field of medical imaging. CNN has demonstrated remarkable success in various computer vision tasks, and we aim to harness their potential to improve medical diagnostics and healthcare outcomes. In the project, we reduced the likelihood of human errors in medical image analysis by automating routine tasks. The purpose of this project is to improve 'Medical Imaging' in the college in real-time.

Key words: Image Processing, Medical Imaging, CNN

"Where the vision is one year,

cultivate flowers;

Where the vision is ten years,

cultivate trees;

Where the vision is eternity,

cultivate people."

- Oriental Saying

List of Figures

List of Tables

List of Abbreviations

Abbr1: R- CNN- Regional based Convolutional Neural Networks

Abbr2: CNN – Convolutional Neural Networks

Abbr3: CADx- Computer aided detection

Abbr4: CAD- Computer aided diagnosis

Abbr5: AMD- age-related macular degeneration

Abbr6: GPU- Graphical Processing Unit

Abbr7: MRI- Magnetic Resonance Imaging

Table of Contents

Chapter 1
Introduction

The modern world is enclosed with gigantic masses of digital visual information. To analyze and organize these devastating oceans of visual information, image analysis techniques are a major requisite. In particular, Medical image classification plays an essential role in clinical treatment and teaching tasks. It has emerged as one of the top research areas in the field of engineering and medicine. Recent years have witnessed rapid use of machine learning algorithms in medical image analysis.The application of Convolutional Neural Networks in medical imaging has the potential to revolutionize healthcare by improving diagnostic accuracy, reducing human error, and enhancing patient care.

The project uses image analysis for object detection and recognition in order to determine the activities and track objects. Here, Machine learning is used to train the machine for a particular set of images so that a system can be implemented for the application of Convolutional Neural Networks (CNNs) in the field of medical imaging.

1.1 Overview

The project is based on image processing and analysis from medical images that are captured by the doctors. Images like ultrasound images, ct scans, x-rays etc which are useful for analysis of various problems and diseases are used in this project.A system is made in such a way that it detects the medical images, classifies the images and stores them in the database. Along with storing the information in the database, it is also able to identify a medical image using the data previously stored in the database.

1.2 Background and Motivation

The deep neural network is an emerging machine learning method that has proven its potential for different classification tasks. Notably, the convolutional neural network dominates with the best results on varying image classification tasks. However, medical image datasets are hard to collect because it needs a lot of professional expertise to label them. The recent advancements in hardware design, safety procedures, computational resources and data storage capabilities has greatly benefited the field of medical imaging. Currently, the major application areas of medical image analysis involve segmentation, classification, and abnormality detection in images generated from a wide spectrum of clinical imaging modalities.

1.3 Problem Statement and Objectives

The field of medical imaging is plagued by challenges that hinder the efficiency and accuracy of disease diagnosis and patient care. Manual interpretation of medical images is time-consuming, prone to human error, and often constrained by the availability of specialized healthcare professionals. To address these critical issues, this project seeks to harness the potential of Convolutional Neural Networks (CNNs) to create an innovative and reliable solution for automated medical image analysis. The problem at hand is to develop, implement, and evaluate CNN-based systems capable of accurately detecting diseases and abnormalities in a variety of medical images, while ensuring data privacy, interpretability, and seamless integration into existing healthcare workflows.

1.4 Scope of the Project

As the project uses image processing for detecting medical images in real-time, it can have a wide variety of applications in various areas. Some of them are given below :

Disease Detection and Diagnosis:

Cancer Detection: CNNs are used to identify tumors and lesions in various types of medical images, including mammograms, CT scans, and MRIs.

Diabetic Retinopathy Detection: CNNs analyze retinal images to identify the early signs of diabetic retinopathy, helping prevent vision loss in diabetes patients.

Cardiovascular Disease Diagnosis: CNNs assist in the detection of heart diseases by analyzing cardiac images and identifying anomalies in the heart structure.

Ophthalmology:

Glaucoma and Cataract Detection: CNNs analyze eye images to identify glaucoma or cataracts, enabling early intervention.

Retinal Disease Diagnosis: Retinal scans are examined to detect diseases like age-related macular degeneration (AMD).

Virtual Endoscopy: CNNs are used for simulating endoscopic procedures, allowing medical professionals to practice and refine their skills in a risk-free virtual environment.

3D Medical Imaging: CNNs are employed in the reconstruction and analysis of 3D medical images, such as 3D CT scans and MRIs, for complex surgeries and treatment planning.

1.5 Team organization

Aanya Chourasiya: I was responsible for the work in the frontend and used languages such as HTML, CSS and JavaScript to design the frontend.
Abdul Rehman: I was responsible for the documentation and the development of the design of the HTML in the frontend of the project.
Amisha Prajapati: I created Use Case and ER Diagram.
Atharva Puranik: I was responsible for the backend of the project. I used machine learning algorithms in flask for the detection of various diseases.

1.6 Report Structure

The project **Medical Imaging using Convolutional Neural Networks** is primarily concerned with the Image processing in real-time and whole project report is categorized into five chapters.

Chapter 1: Introduction - introduces the background of the problem followed by rationale for the project undertaken. The chapter describes the objectives, scope and applications of the project. Further, the chapter give the details of team members and their contribution in development of project which is then subsequently ended with report outline.

Chapter 2: Review of Literature - explores the work done in the area of Project undertaken and discusses the limitations of the existing system and highlights the issues and challenges of the project area. The chapter finally ends up with the requirement identification for present project work based on findings drawn from reviewed literature and end user interactions.

Chapter 3: Medical Imaging Using CNN - starts with the project proposal based on requirement identified, followed by benefits of the project. The chapter also illustrates the software engineering paradigm used along with different design representations. The chapter also includes a block diagram and details of major modules of the project. Chapter also gives insights of different types of feasibility study carried out for the project undertaken. Later it gives details of the different deployment requirements for the developed project.

Chapter 4: Implementation - includes the details of different Technology/ Techniques/ Tools/ Programming Languages used in developing the Project. The chapter also includes the different user interfaces designed in the project along with their functionality. Further it discusses the experiment results along with testing of the project. The chapter ends with evaluation of the project on different parameters like accuracy and efficiency.

Chapter 5: Conclusion - Concludes with objective wise analysis of results and limitation of present work which is then followed by suggestions and recommendations for further improvement.

Chapter 2
Review of Literature

Medical image classification is a sub-subject of image classification. Many techniques in image classification can also be used on it. Such as many image enhanced methods to enhance the discriminable features for classification. Medical image analysis aims to aid radiologists and clinicians to make the diagnostic and treatment process more efficient.

The computer aided detection (CADx) and computer aided diagnosis (CAD) relies on affective medical image analysis making it crucial in terms of performance, since it would directly affect the process of clinical diagnosis and treatment. Therefore, high performance in terms of accuracy, precision, recall, sensitivity, and specificity is crucial and most desirable in medical image analysis.

2.1 Preliminary Investigation

2.1.1 Current System

Google's DeepMind Health:

Google's DeepMind Health utilizes machine learning algorithms for medical image analysis. Their system employs convolutional neural networks (CNNs) and deep learning techniques to assist in the analysis of medical imaging data, such as MRI (Magnetic Resonance Imaging) and CT (Computed Tomography) scans. DeepMind Health aims to improve diagnostic accuracy, detect abnormalities, and assist healthcare professionals in interpreting complex medical images more efficiently.

IBM Watson Health Imaging:

IBM Watson Health offers AI-powered imaging solutions that leverage machine learning algorithms for medical image analysis. Using deep learning models, Watson Health Imaging assists in analyzing radiological images, including X-rays and mammograms. The system aims to aid radiologists in detecting anomalies, identifying patterns, and providing more accurate diagnoses by augmenting human expertise with AI-driven insights.

GE Healthcare's Edison AI Platform:

GE Healthcare's Edison platform integrates artificial intelligence and machine learning into medical imaging technologies. The platform incorporates various AI-driven applications, including image reconstruction, segmentation, and analysis. Edison uses machine learning algorithms to enhance image quality, assist in disease detection, and streamline radiology workflows. It aims to improve diagnostic accuracy and efficiency in interpreting medical images like MRI, CT, and ultrasound scan.

2.2 Limitations of Current System

Google's DeepMind Health:

- Limited integration into clinical settings due to regulatory and privacy concerns regarding patient data.
- Challenges in ensuring interoperability with existing healthcare systems and workflows.

IBM Watson Health Imaging:

- Issues with the reproducibility and generalization of AI models across diverse patient populations and imaging equipment.
- Concerns regarding the complexity of integrating AI solutions into existing radiology workflows.

GE Healthcare's Edison AI Platform:

- Difficulty in achieving seamless integration of AI tools with different imaging modalities and vendor-specific systems.
- Challenges related to the interpretability and transparency of AI-driven insights for radiologists and healthcare practitioners.

2.3 Requirement Identification and Analysis for Project

Significant work has been done in the field of Medical Imaging using Convolutional Neural Networks; however, it is not easy to achieve desired results. The review of literature leads to draw certain major findings which are as under :

This study brought out how to apply the convolutional neural network (CNN) based algorithm on a chest X-ray dataset to classify pneumonia. Three techniques are evaluated through experiments. These are linear support vector machine classifiers with local rotation and orientation free features, transfer learning on two convolutional neural network models: Visual Geometry Group i.e., VGG16 and InceptionV3, and a capsule network training from scratch. [1]

Among deep learning techniques, deep convolutional networks are actively used for the purpose of medical image analysis. This includes application areas such as segmentation, abnormality detection, disease classification, computer aided diagnosis and retrieval. In this study, a comprehensive review of the current state-of-the-art in medical image analysis using deep convolutional networks is presented. The challenges and potential of these techniques are also highlighted. [2]

This article aims to provide a comprehensive survey of applications of CNNs in medical image understanding. The underlying objective is to motivate medical image understanding researchers to extensively apply CNNs in their research and diagnosis. A brief introduction to CNNs has been presented. A discussion on CNN and its various award-winning frameworks have been presented. The major medical image understanding tasks, namely image classification, segmentation, localization and detection have been introduced.[3]

In this paper, the author provides a survey on convolutional neural networks in medical image analysis. First, we review the commonly used CNNs in medical image processing, including AlexNet, GoogleNet, ResNet, R-CNN, and F-CNN. Then, we present an overview of the use of CNNs, for image classification, segmentation, detection, and other tasks such as registration, content-based image retrieval, image generation. [4]

This paper provides detailed analysis of deep learning and its techniques used in various applications and especially to provide an extensive reference for the researchers in deep learning and its algorithms, implementation techniques and applications used in recent technologies. This paper will also help to improve investigation of deep learning and highlights new research areas and advancements of technology. [5]

The biomedical picture classification aims to identify and classify biomedical characteristics efficiently, which have significant advantages to numerous study and development fields. In this paper, the framework focused on the different architectures that were used to classify the medical images along with their performances. [6]

In this paper, we investigated two major directions for explaining convolutional neural networks: feature-based post hoc explanatory methods that try to explain already trained and fixed target models and preliminary analysis and choice of the model architecture with an accuracy of 98% ± 0.156% from 36 CNN architectures with different configurations. [7]

The authors begin the survey by providing a synopsis of research works in medical imaging based on convolutional neural networks. Second, we discuss popular pretrained models and general adversarial networks that aid in improving convolutional networks' performance. [8]

This review introduces a few popular algorithms using Convolutional Neural Networks(CNNs) being used in the field along with their applications: Classification, Detection, Segmentation, Registration and Image Enhancement. The paper further provides some useful resources on some of the most promising anatomical areas of application in medical image analysis with Convolutional Neural Networks: brain, breast, chest, eye and skin. [9]

2.3.1 Conclusion

This chapter reviews the literature surveys that have been done during the research work. The related work that has been proposed by many researchers has been discussed. The research papers related to object detection and recognition of objects from 1985 to 2015 have been shown which discussed different methods and algorithm to identify objects.

Chapter 3
Medical Imaging using CNN

3.1 The Proposal

The proposal is to create a system which can identify various medical images to help the detection of diseases like brain tumor, lung diseases and help doctors to identify and deal with them.

It can also classify the images if it is brains, muscles, lungs, liver etc. and store that information in the database. It can give the results by analyzing different images and produce results by the information in the database.

3.2 Benefits of the Proposed System

The current system had a lot of challenges that are overcome by this system:

Accurate Diagnostics: CNNs can provide highly accurate and consistent diagnostic assessments. They can detect subtle patterns and anomalies in medical images that might be missed by human observers.

Faster Diagnoses: CNNs can process medical images rapidly, reducing the time required for analysis. This can lead to quicker diagnoses and more timely interventions.

Reduced Human Error: Automation through CNNs can significantly reduce human error in image interpretation, improving the reliability of diagnoses.

Access to Expertise: CNNs can act as a "second opinion" tool, especially in regions with limited access to specialized healthcare professionals, providing access to expertise remotely.

3.3 Block Diagram of Medical Diagnosis

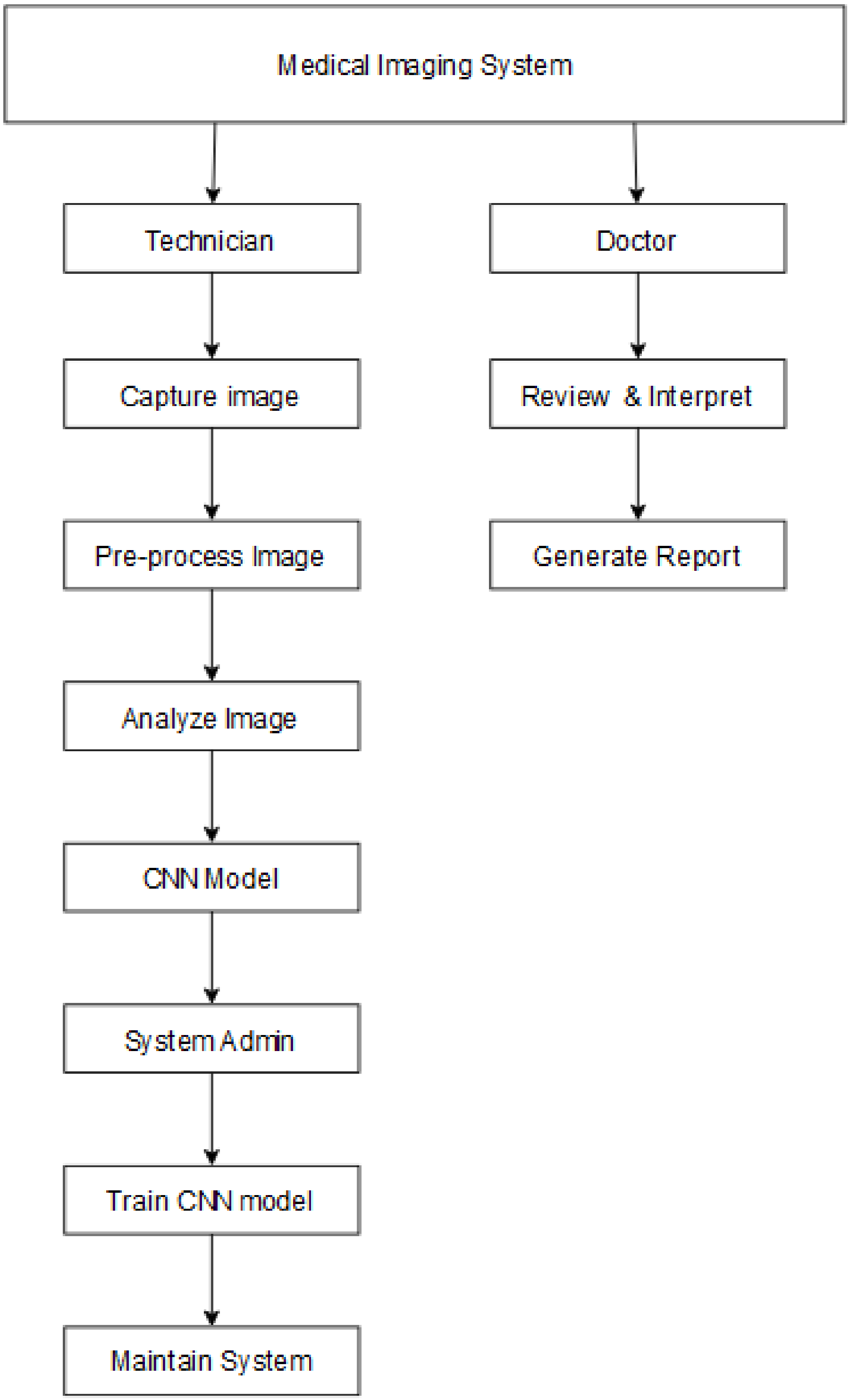

Figure 3-1 : Block Diagram of Medical Diagnosis

3.4 Feasibility Study

A feasibility study is an analysis of how successfully a system can be implemented, accounting for factors that affect it such as economic, technical and operational factors to determine its potential positive and negative outcomes before investing a considerable amount of time and money into it.

3.4.1 Technical

For any real-time detection system, there is a need to process images from the video. For this, the kind of framework used must be the one that is capable of extracting those objects from the images easily and accurately in real-time. The framework used in this is Flask, which is a framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries.The system once set up completely, works automatically with high performance and efficiency. The result (count and other information), gets automatically saved in the database.

3.4.2 Economical

For any real-time object detection system, there is a need for a High definition Camera for better and accurate results.Since the system is completely automated, there is a need for continuous electricity supply for it to operate 24x7.The Flask framework used in the system works great with GPU built systems, which are a little on the expensive side. Since the system uses high performance processors continuously, to save any disaster from occurring due to very high temperatures, there is a requirement of a cooling system in the environment where it is implemented.

3.4.3 Operational

The main motto of our system is to reduce the manual efforts of counting the students and vehicles by automating it. The system is able to do that accurately and efficiently making the system operationally feasible.

3.5 Design Representation

3.5.1 Data Flow Diagrams

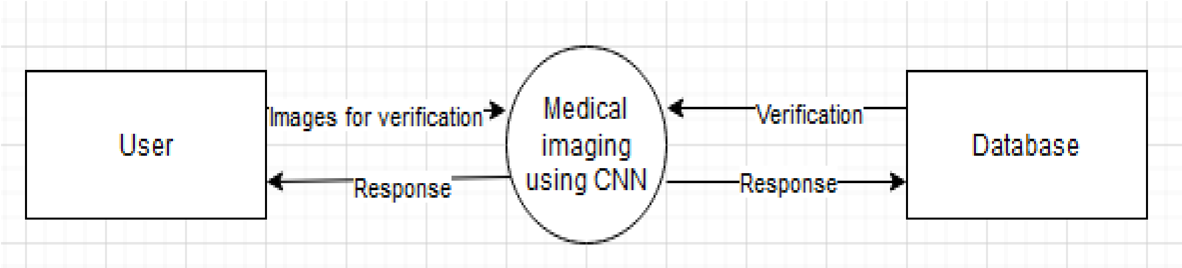

Figure 3-2: Data Flow Diagram Level 0 of medical imaging

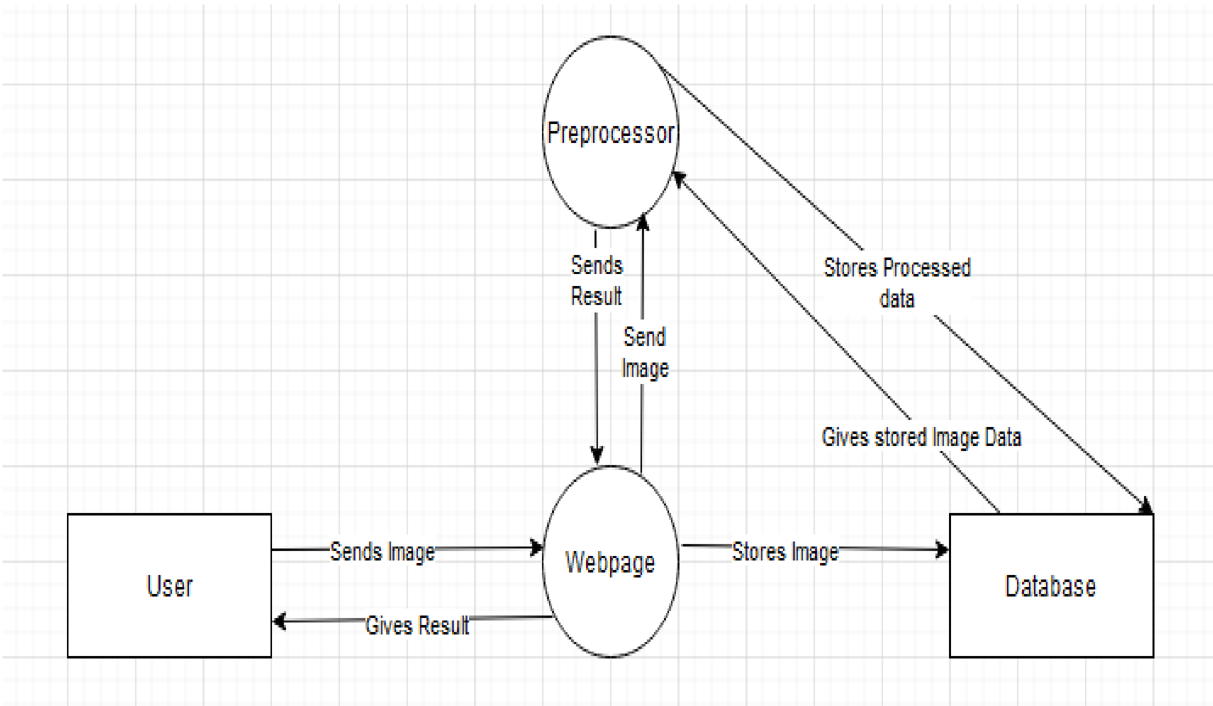

Figure 3-3: Data Flow Diagram Level 1 of medical imaging

3.5.2 Use case diagram

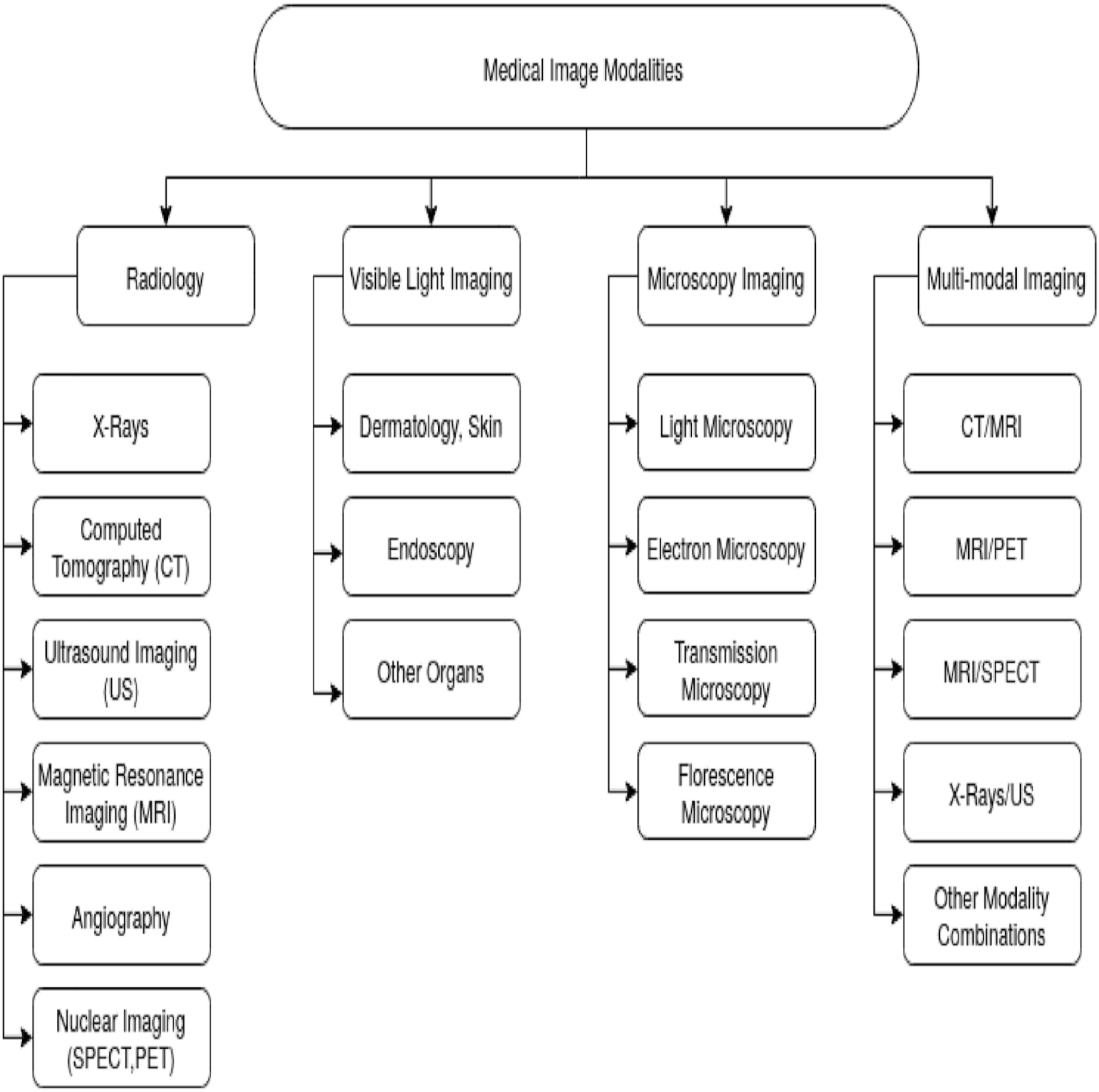

Figure 3-4: Use-case diagram of medical imaging

3.5.3 Entity relationship diagram

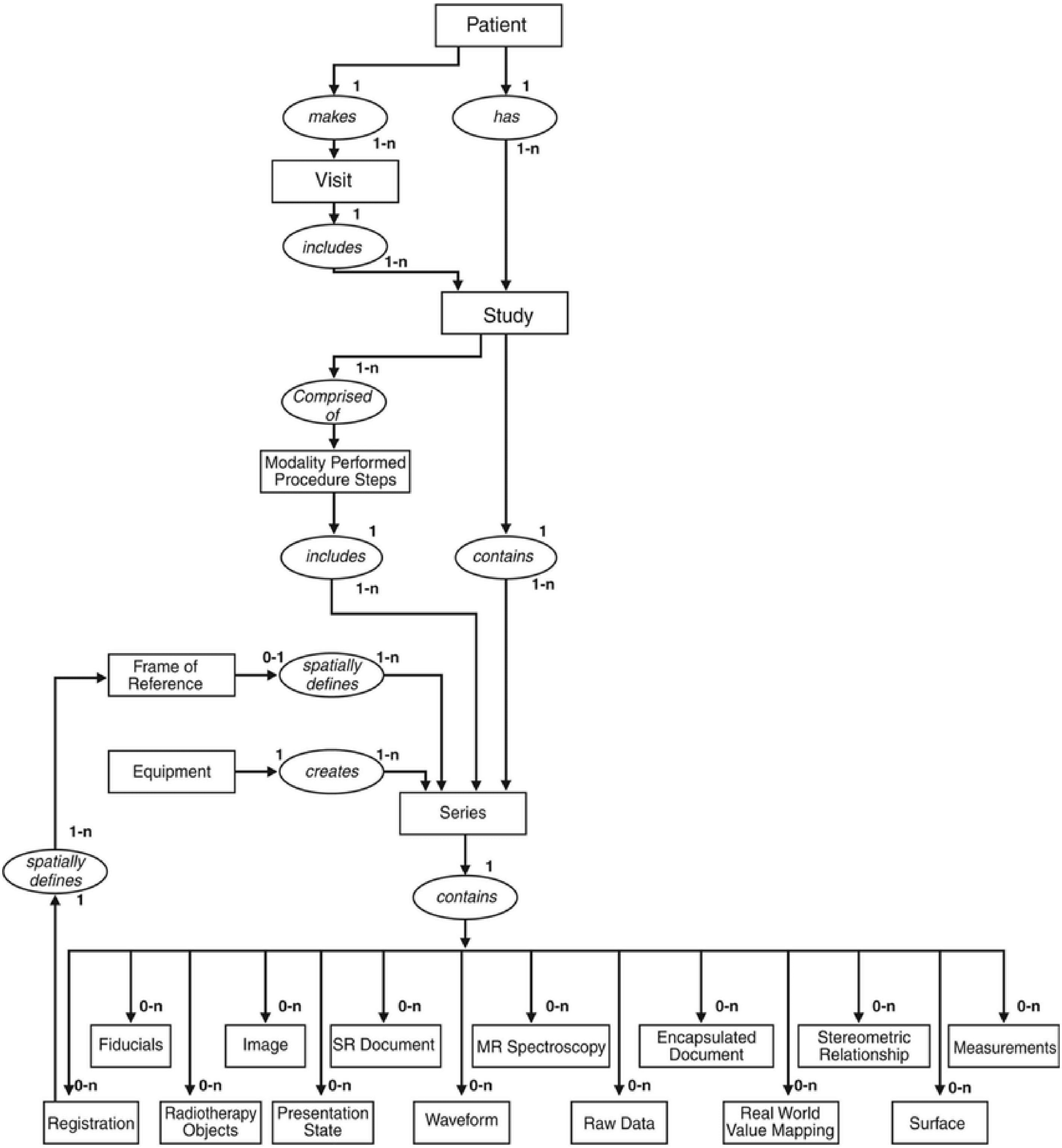

Figure 3-5: E-R Diagram of medical imaging

3.6 Deployment Requirements

There are various requirements (hardware, software and services) to successfully deploy the system. These are mentioned below :

3.6.1 Hardware

32-bit or 64-bit Processing System: Refers to the type of processor architecture the system should have, either 32-bit or 64-bit.
Operating System: Windows 7 or later is required for the software to run properly.
Computer System: High processing power is recommended. Optionally, a GPU (Graphical Processing Unit) is suggested for enhanced performance.

3.6.2 Software

Python: A programming language used for developing applications, including machine learning and web-based projects.
Flask: A web framework for Python. Flask is used to build web applications, including web-based interfaces for machine learning models or other software systems.

Chapter 4
Implementation

For the problem of identifying and classifying medical images, the system is designed in such a way so as to take the image and verify the various medical problems which are shown in the image using the data present in the database.

4.1 Technique Used

4.1.1 Deep- Learning :

Deep Learning is a subfield of machine learning concerned with algorithms inspired by the structure and function of the brain called artificial neural networks. Deep learning (also known as deep structured learning or hierarchical learning) is part of a broader family of machine learning methods based on learning data representations, as opposed to task-specific algorithms. Learning can be supervised, semi-supervised or unsupervised.Deep learning models are loosely related to information processing and communication patterns in a biological nervous system, such as neural coding that attempts to define a relationship between various stimuli and associated neuronal responses in the brain.

Deep learning architectures such as deep neural networks, deep belief networks and recurrent neural networks have been applied to fields including computer vision, speech recognition, natural language processing, audio recognition, social network filtering, machine translation, bioinformatics and drug design, where they have produced results comparable to and in some cases superior to human experts.

4.1.2 Neural Networks :

In machine learning, a convolutional neural network (CNN or ConvNet) is a class of deep, feed-forward artificial neural networks that has successfully been applied to analyzing visual imagery.

CNNs use a variation of multilayer perceptrons designed to require minimal preprocessing. They are also known as shift invariant or space invariant artificial neural networks (SIANN), based on their shared-weights architecture and translation invariance characteristics.

Convolutional networks were inspired by biological processes in that the connectivity pattern between neurons resembles the organization of the animal visual cortex. Individual cortical neurons respond to stimuli only in a restricted region of the visual field known as the receptive field. The receptive fields of different neurons partially overlap such that they cover the entire visual field. CNNs use relatively little pre-processing compared to other image classification algorithms. This means that the network learns the filters that in traditional algorithms were hand-engineered. This independence from prior knowledge and human effort in feature design is a major advantage. They have applications in image and video recognition, recommender systems and natural language processing.

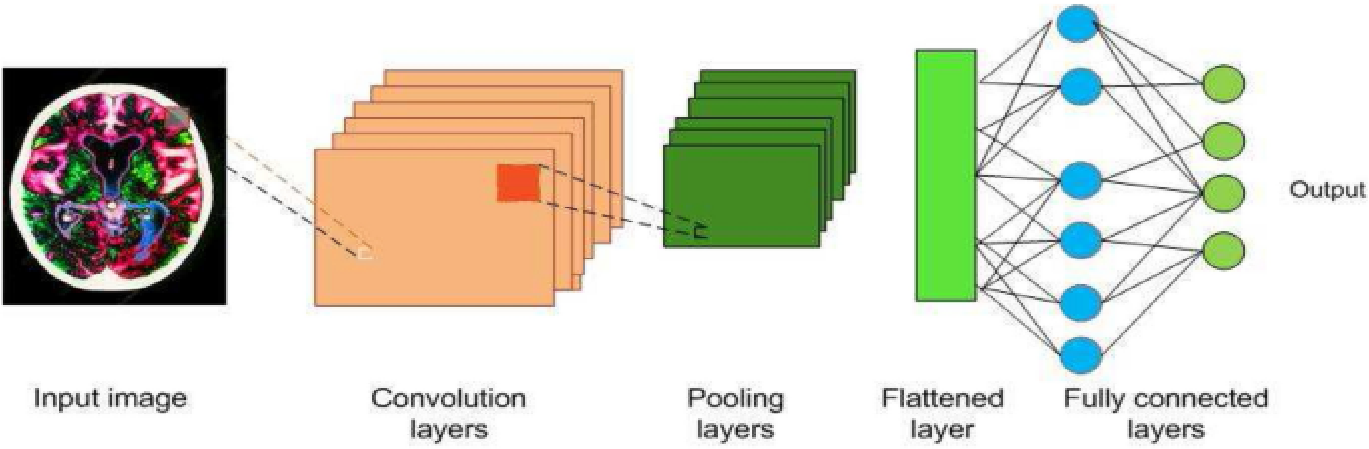

Figure 4-1 : Neural Networks
[The Figure shows a collection of statistical machine learning techniques used to learn feature hierarchies often based on artificial neural networks]

4.2 Tools Used

4.2.1 Flask

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools.

4.2.2 OpenCV

OpenCV (Open Source Computer Vision Library) is released under a BSD license and hence it's free for both academic and commercial use. It has C++, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android. OpenCV was designed for computational efficiency and with a strong focus on real-time applications. Written in optimized C/C++, the library can take advantage of multi-core processing. Enabled with OpenCL, it can take advantage of the hardware acceleration of the underlying heterogeneous compute platform. Adopted all around the world, OpenCV has more than 47 thousand users and an estimated number of downloads exceeding 14 million. Usage ranges from interactive art, to mine inspection, stitching maps on the web or through advanced robotics.

4.2.3 TensorFlow

TensorFlow is a versatile and powerful open-source software library designed for high-performance numerical computation. Its adaptability allows the seamless deployment of computations across a diverse range of platforms, from traditional central processing units (CPUs) to graphics processing units (GPUs) and even Tensor Processing Units (TPUs). This scalability extends from desktop computers to expansive clusters of servers and, notably, to mobile and edge devices, making it an indispensable tool in today's computing landscape. Originally conceived and developed by the proficient researchers and engineers of the Google Brain team within Google's AI organization, TensorFlow has gained widespread recognition and adoption due to its robust support for machine learning and deep learning applications. Furthermore, its flexible numerical computation core is not limited to the realm of AI but extends its utility to numerous scientific domains, showcasing its significance in various scientific and computational fields.

4.3 Tech Stack

Python

Python is a simple and minimalistic language. Reading a good Python program feels almost like reading English (but very strict English!). This pseudocode nature of Python is one of its greatest strengths. It allows you to concentrate on the solution to the problem rather than the syntax i.e. the language itself. Python supports procedure-oriented programming as well as object-oriented programming. In procedure-oriented languages, the program is built around procedures or functions which are nothing but reusable pieces of programs. In object-oriented languages, the program is built around objects which combine data and functionality. Python has a very powerful but simple way of doing object-oriented programming, especially, when compared to languages like C++ or Java.

HTML

The HyperText Markup Language or HTML is the standard markup language for documents designed to be displayed in a web browser. It defines the meaning and structure of web content. It is often assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript. Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for its appearance. HTML can embed programs written in a scripting language such as JavaScript, which affects the behavior and content of web pages. The inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), former maintainer of the HTML and current maintainer of the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.

CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML or XML (including XML dialects such as SVG, MathML or XHTML). CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript. CSS is designed to enable the separation of content and presentation, including layout, colors, and fonts. This separation can improve content accessibility; provide more flexibility and control in the specification of presentation characteristics; enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, which reduces complexity and repetition in the structural content; and enable the .css file to be cached to improve the page load speed between the pages that share the file and its formatting.

Javascript

JavaScript, often abbreviated as JS, is a programming language that is one of the core technologies of the World Wide Web, alongside HTML and CSS. As of 2023, 98.7% of websites use JavaScript on the client side for webpage behavior, often incorporating third-party libraries. All major web browsers have a dedicated JavaScript engine to execute the code on users' devices. JavaScript is a high-level, often just-in-time compiled language that conforms to the ECMAScript standard. It has dynamic typing, prototype-based object-orientation, and first-class functions. It is multi-paradigm, supporting event-driven, functional, and imperative programming styles. It has application programming interfaces (APIs) for working with text, dates, regular expressions, standard data structures, and the Document Object Model (DOM).

4.4 Visualization

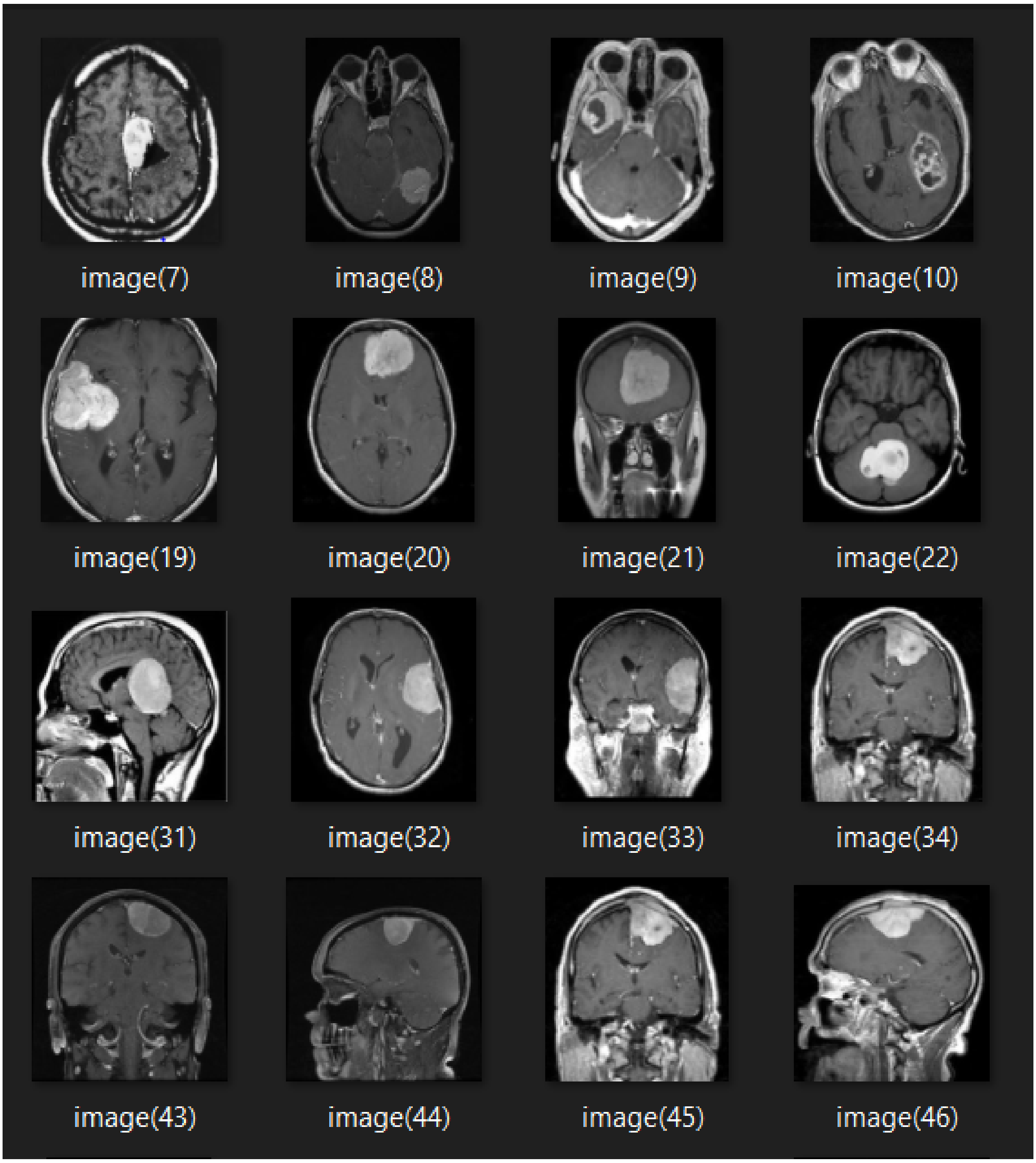

Figure 4-2 : Meningioma Tumor training datasets

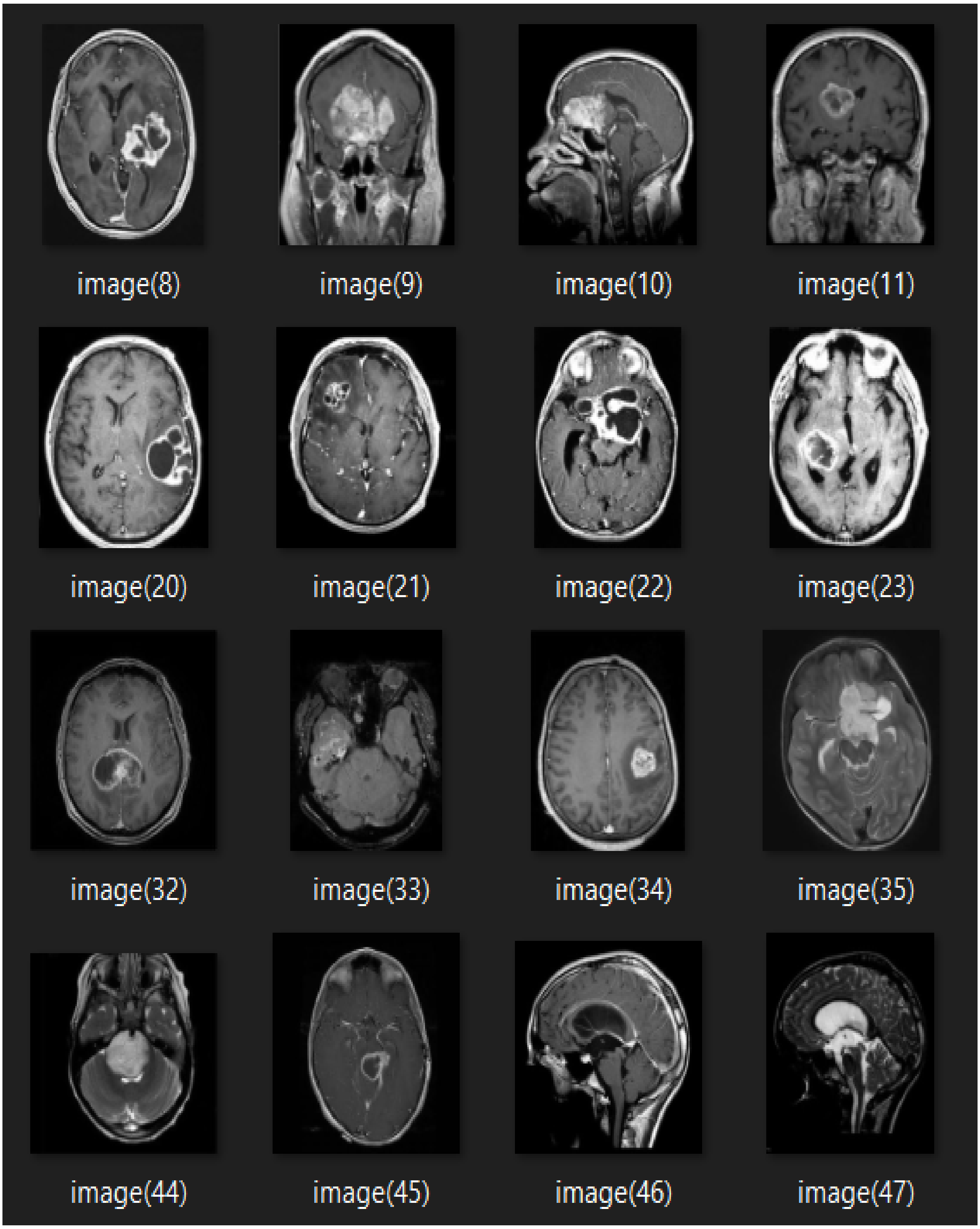

Figure 4-3 : Glioma Tumor training datasets

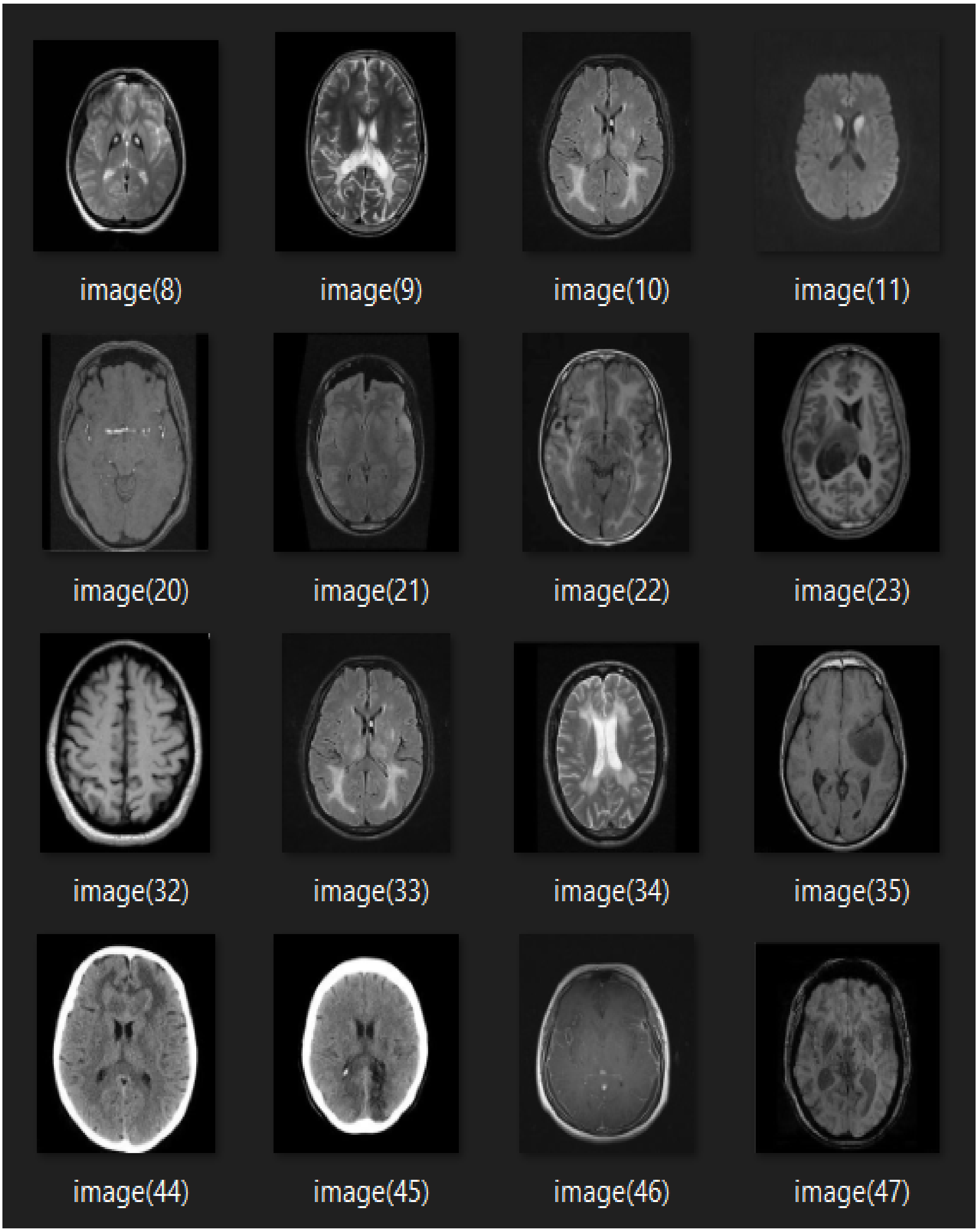

Figure 4-4 : No Tumor training dataset

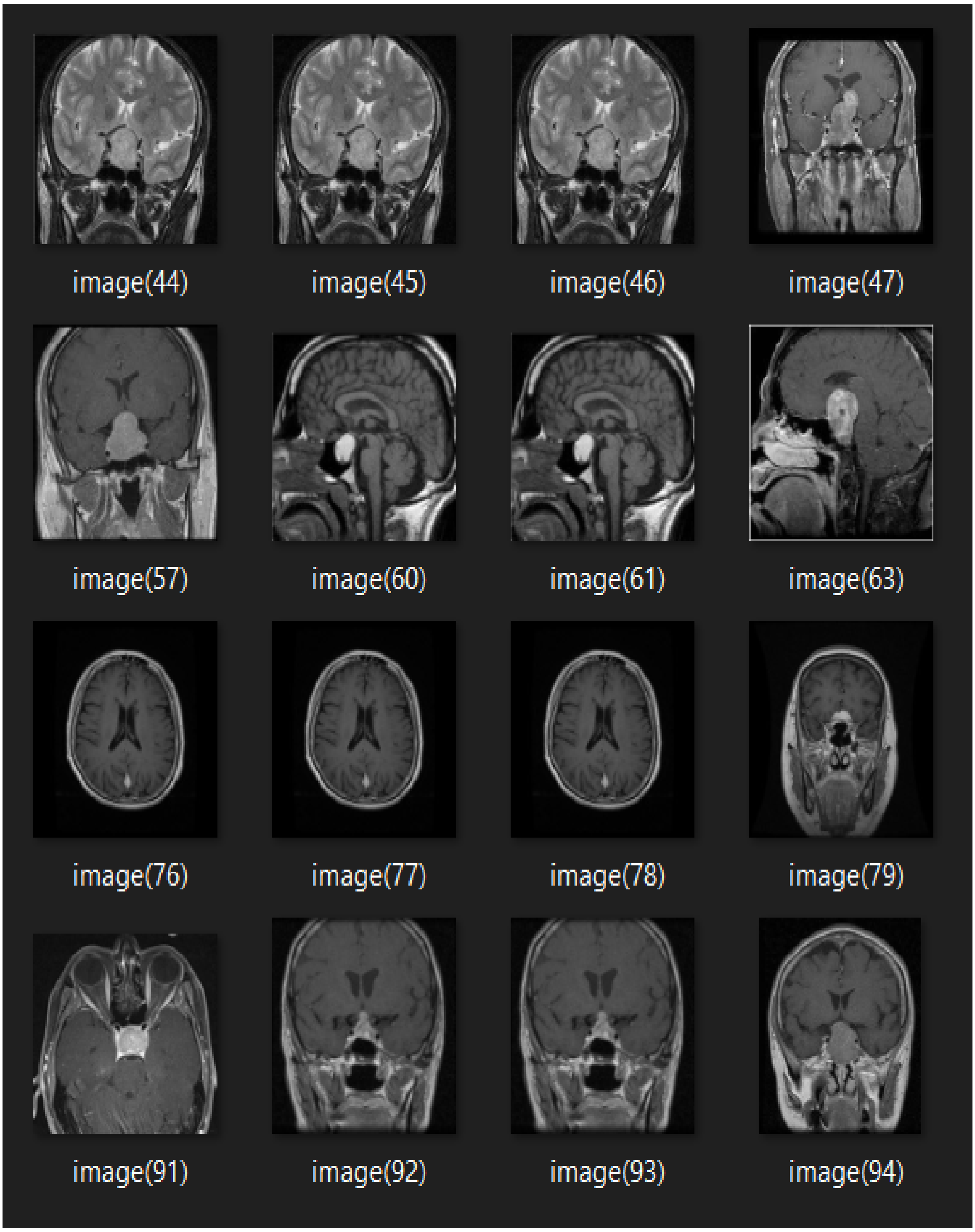

Figure 4-5 : Pituitary Tumor training dataset

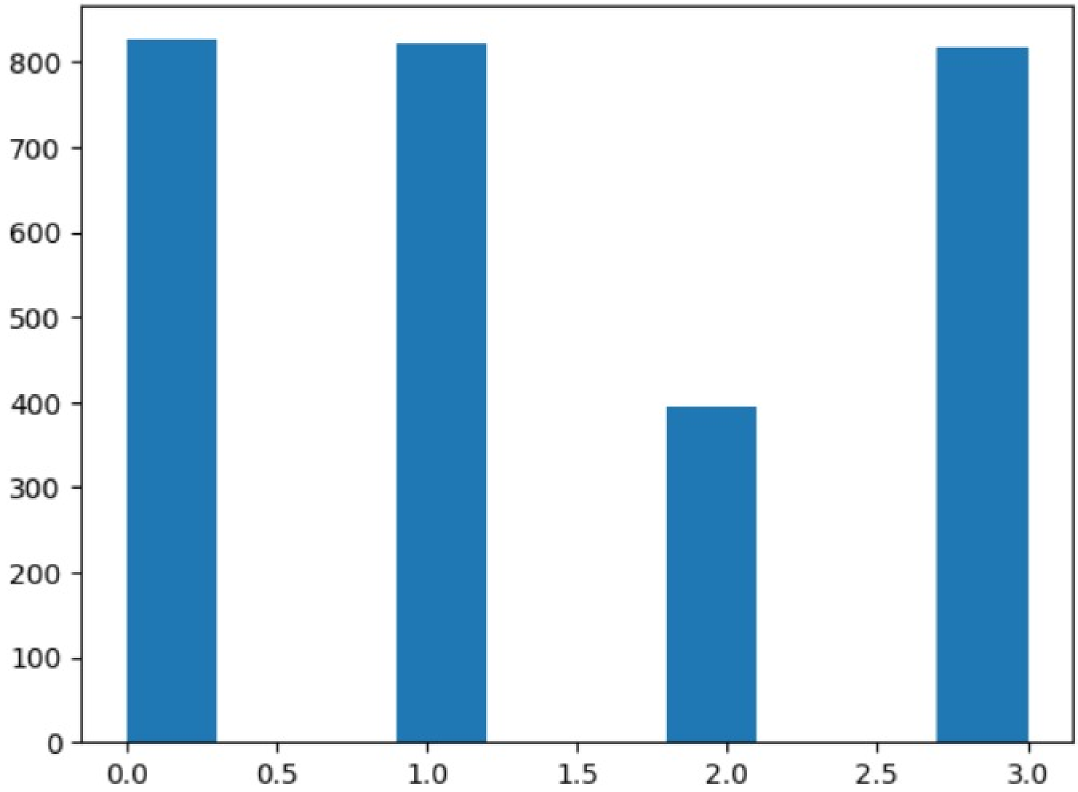

Figure 4-6 : Class Frequency Distribution

[Graph demonstrating the 4 classes i.e meningioma tumor, glioma tumor, no tumor and pituitary tumor]

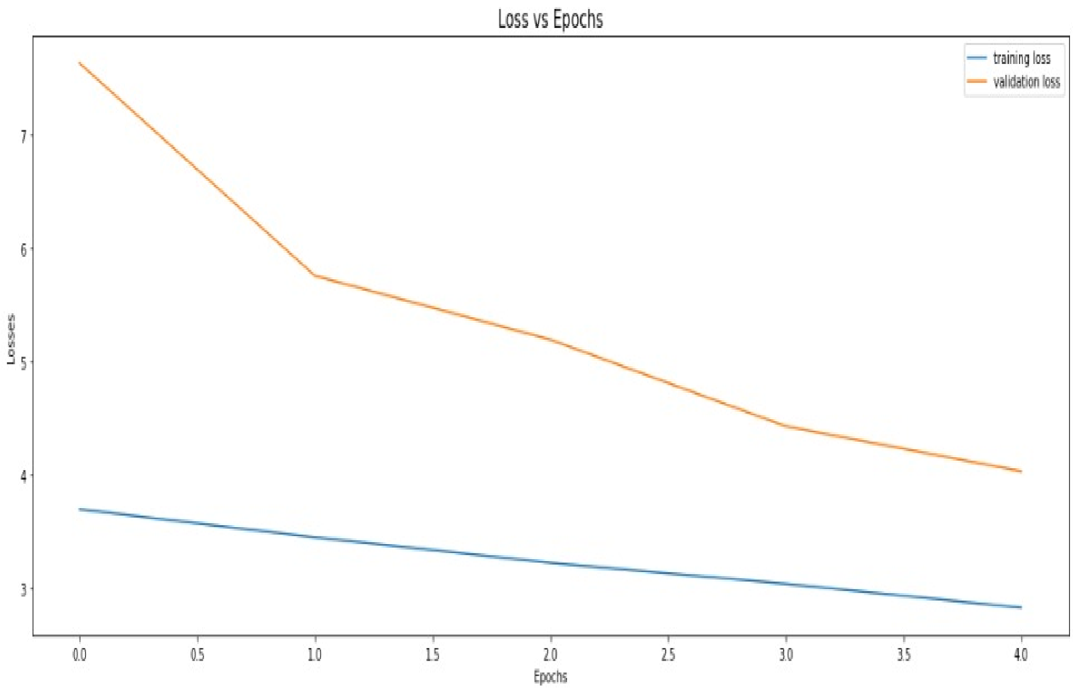

Figure 4-7 : Loss vs Epoch (5 epochs)

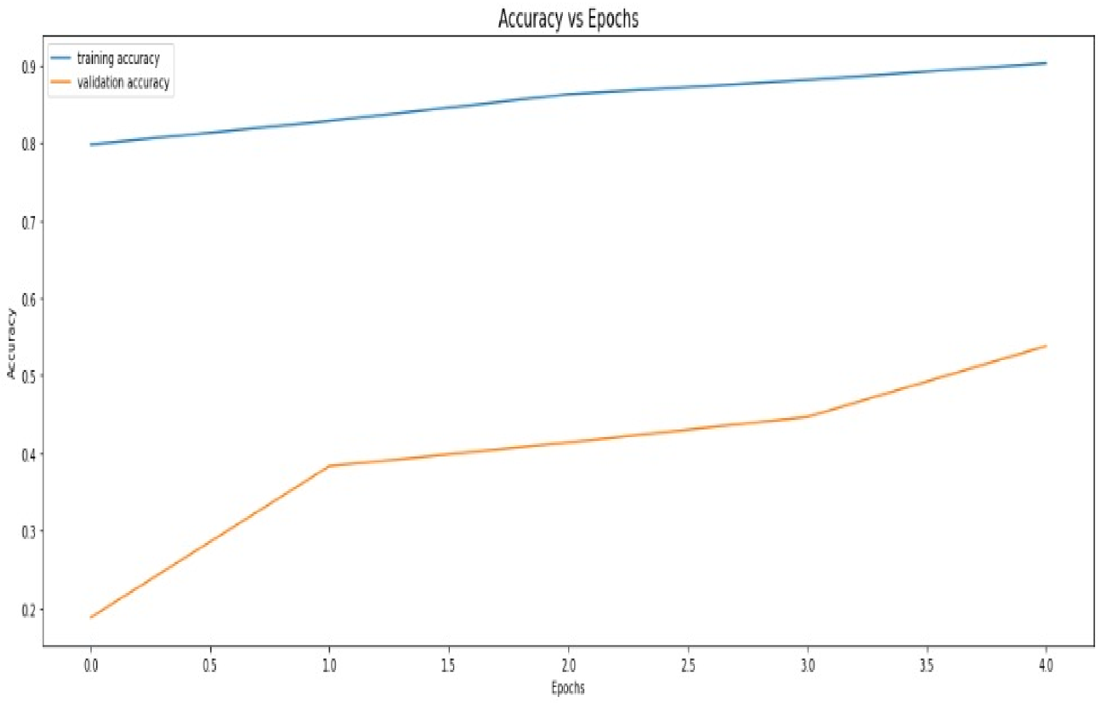

Figure 4-8 : Accuracy vs Epoch (5 epochs)

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 sequential_1 (Sequential)   (None, 100, 100, 128)     1792

 sequential_2 (Sequential)   (None, 50, 50, 64)        74048

 sequential_3 (Sequential)   (None, 25, 25, 32)        18592

 flatten (Flatten)           (None, 20000)             0

 sequential_4 (Sequential)   (None, 128)               2560640

 sequential_5 (Sequential)   (None, 64)                8512

 dense_2 (Dense)             (None, 4)                 260

=================================================================
Total params: 2663844 (10.16 MB)
Trainable params: 2663012 (10.16 MB)
Non-trainable params: 832 (3.25 KB)
_________________________________________________________________
None
```

Figure 4-9 : CNN Model Summary

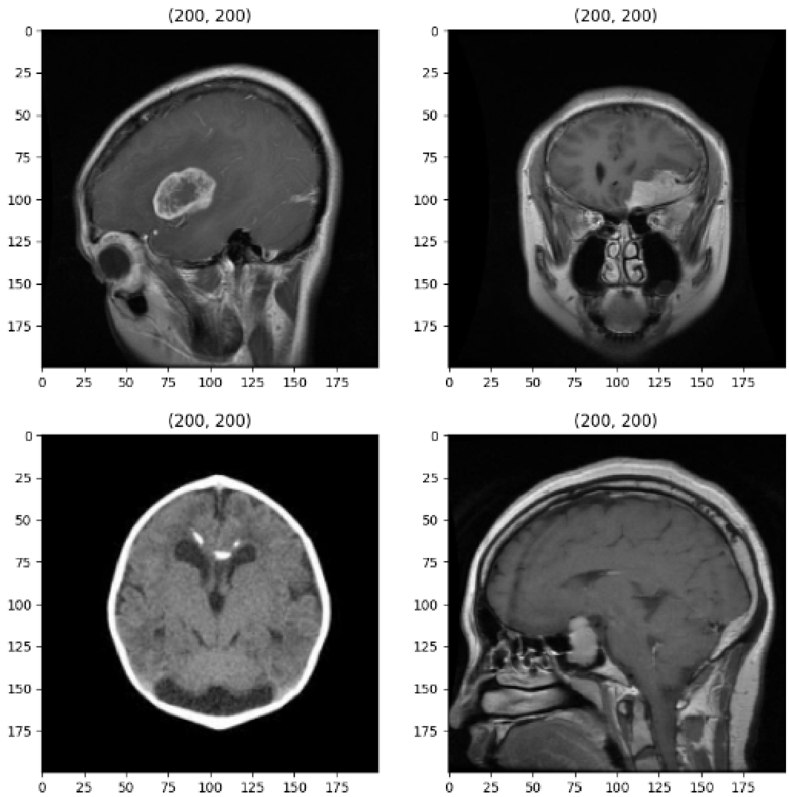

Figure 4-10 : Displaying Image Array
[An array of images representing the datasets]

4.5 Testing

Testing is the process of evaluation of a system to detect differences between given input and expected output and also to assess the feature of the system. Testing assesses the quality of the product. It is a process that is done during the development process.

4.5.1 Strategy Used

Tests can be conducted based on two approaches –

- Functionality testing
- Implementation testing

The texting method used here is Black Box Testing. It is carried out to test functionality of the program. It is also called 'Behavioral' testing. The tester in this case, has a set of input values and respective desired results. On providing input, if the output matches with the desired results, the program is tested 'ok', and problematic otherwise.

4.5.2 Test Case and Analysis

TEST CASE: 1

Test Case ID	TC001
Test Case Summary	To check whether the data uploaded is processed and stored in the database
Test Procedure	Upload the image in the website
Expected Result	Data must be uploaded in the website and stored in database
Actual Result	The data is stored in the database
Status	Pass

Table 1:Test Case 1

TEST CASE 1 OUTPUT :

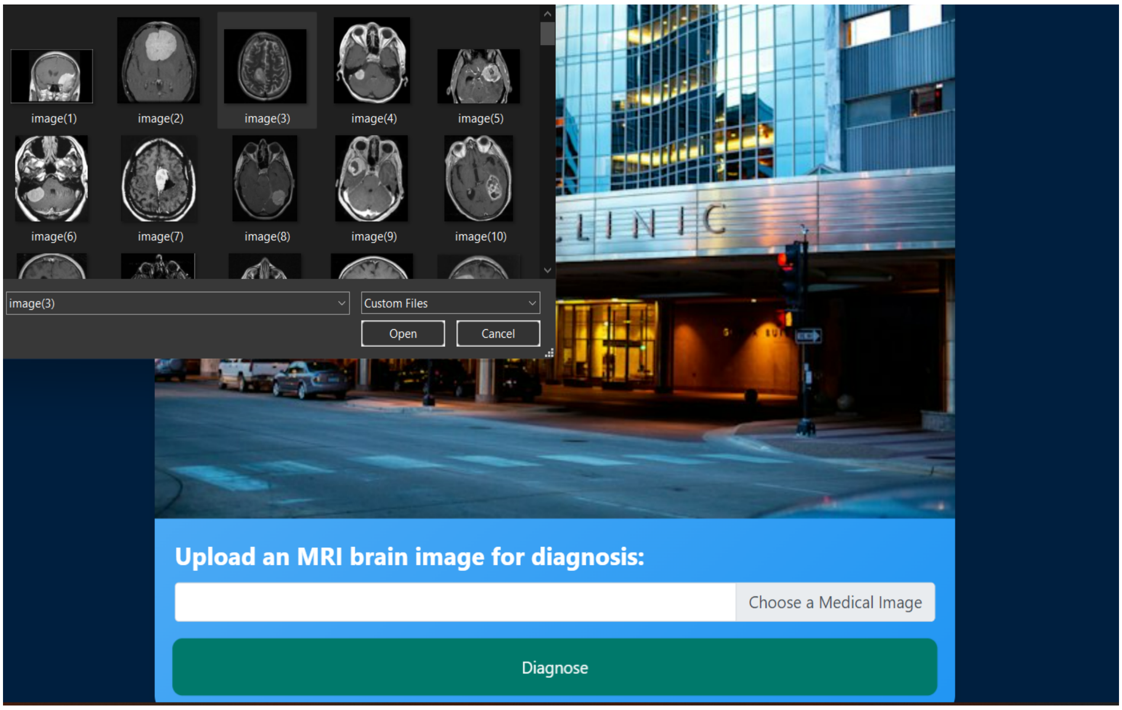

Figure 4-11 : Test Case 1
[Uploading the image in the website]

TEST CASE: 2

Test Case ID	TC002
Test Case Summary	To check whether the system is able to detect tumor or not
Test Procedure	Upload the image in the website
Expected Result	The website will be able to detect tumor on the image
Actual Result	The website returns information about which tumor the brain has.
Status	Pass

Table 2: Test Case 2

TEST CASE 2 OUTPUT

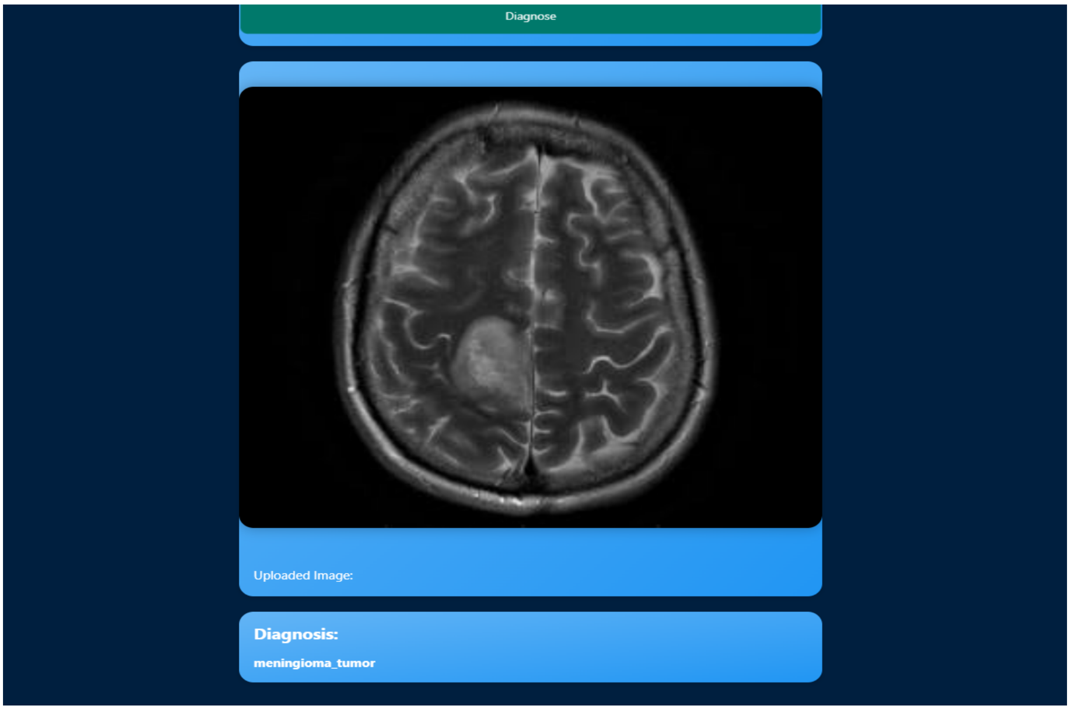

Figure 4-12 : Test Case 2
[Predicting the output]

Chapter 5
Conclusion

5.1 Conclusion

The aim of the project that was to automatically detect, identify various medical images of various organs and human body parts and save their details in the database is successfully and accurately done by this project with the use of concepts like Deep learning, Convolutional Neural Networks, OpenCV and TensorFlow. The work done manually can now be completely replaced by this automated system and it can reduce all the extra efforts of maintaining the records.

5.2 Limitations of the Work

- The working of this project would be a little slow because framework like TensorFlow, & deep learning need high-processing hardware and GPU(graphical processing unit) systems but we are using CPU only.
- The models that we are using for identifying the objects are pre-trained models. So, if we want to train our own model, it takes a lot of time and processing.
- In the system, scanning of each frame is one per second but still it needs improvement. If the objects move too fast, it may not detect them.

5.3 Suggestion and Recommendations for Future Work

The Model would be trained to Investigate the use of federated learning to enable model training on decentralized healthcare data, preserving data privacy while creating powerful AI models.Develop AI models that can forecast the progression and severity of diseases based on historical patient data and imaging.

Bibliography

[1]Deep convolutional neural network based medical image classification for disease diagnosis,Samir S. Yadav &Shivajirao M. Jadhav

[2]Medical Image Analysis using Convolutional Neural Networks: A Review, Syed Muhammad Anwar, Muhammad Majid, Adnan Qayyum, Muhammad Awais, Majdi Alnowami & Muhammad Khurram Khan

[3]Convolutional neural networks in medical image understanding: a survey, D. R. Sarvamangala and Raghavendra V. Kulkarni

[4]Convolutional neural networks for medical image analysis: State-of-the-art, comparisons, improvement and perspectives, Hang Yu, Laurence T. Yang, Qingchen Zhang, David Armstrong, M. Jamal Deen

[5]Deep Convolutional Neural Networks (CNN) for Medical Image Analysis, Mohana Pundlik Patil, SP. Chokkalingam

[6]Biomedical image classification using deep convolutional neural networks – overview by M Bharath Simha Reddy and Pooja Rana

[7]CXAI: Explaining Convolutional Neural Networks for Medical Imaging Diagnostic Zakaria Rguibi , Abdelmajid Hajami , Dya Zitouni , Amine Elqaraoui and Anas Bedraoui

[8]A Review Paper about Deep Learning for Medical Image Analysis, Bagher Sistaninejhad, Habib Rasi, and Parisa Nayeri

[9]A Review on Medical Image Analysis with Convolutional Neural Networks by Paarth Bir and Valentina E. Balas

Source Code

Model.py

```
import numpy as np
import pandas as pd
import os
import matplotlib.pyplot as plt
import seaborn as sb
sb.set_style('whitegrid')
import tensorflow as tf
from tensorflow.keras import layers, models, optimizers, preprocessing
from tensorflow.keras.layers import Conv2D, Dense, MaxPooling2D, Flatten, Dropout

# from google.colab import drive
# drive.mount('/content/drive')
main_dir = "C:\\Users\\ss\\OneDrive\\Desktop\\Brain Tumor CNN\\Dataset"
classification_dirs = [("no_tumor",), ("meningioma_tumor",), ("glioma_tumor",),
("pituitary_tumor",)]
resolution = 64

def load_images(root_dir_name):
    x = []
    y = []

    for label, sub_dir_names in enumerate(classification_dirs):
        for sub_dir_name in sub_dir_names:
            print(f"loading {root_dir_name} {sub_dir_name}")
            sub_dir_path = os.path.join(main_dir, root_dir_name, sub_dir_name)
            for image_name in os.listdir(sub_dir_path):
                image_path = os.path.join(sub_dir_path, image_name)
                image = preprocessing.image.load_img(image_path, color_mode="grayscale",
target_size=(resolution, resolution))
                x.append(preprocessing.image.img_to_array(image))
                y.append(label)

    x = np.array(x) / 255.0
    y = np.array(y)

    return x, y
```

```
x_train, y_train = load_images("Training")
x_test, y_test = load_images("Testing")

y_train = tf.keras.utils.to_categorical(y_train, num_classes=4)
y_test = tf.keras.utils.to_categorical(y_test, num_classes=4)

c = 10
fig, subplots = plt.subplots(1, c)
fig.set_size_inches(25, 3)
for i in range(c):
    n = np.random.randint(0, len(x_train))
    num = np.argmax(y_train[n])
    word = "out" if num == 0 else ""

    subplots[i].imshow(x_train[n].reshape((resolution, resolution)), cmap="gray")
    subplots[i].set_title(f"brain with{word} tumor: {num}")
    subplots[i].axis("off")
plt.show()

input_shape = (64,64,1)
model = models.Sequential()
model.add(Conv2D(32, kernel_size=(2, 2), strides=(1, 1), activation='relu',
input_shape=input_shape))
model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))
model.add(Conv2D(64, kernel_size=(2, 2), strides=(1, 1), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))
model.add(Conv2D(128, kernel_size=(2, 2), strides=(1, 1), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))
model.add(Conv2D(256, kernel_size=(2, 2), strides=(1, 1), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))
model.add(Conv2D(512, kernel_size=(2, 2), strides=(1, 1), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))
model.add(Flatten())
model.add(Dropout(0.5))
model.add(Dense(256, activation='relu'))
model.add(Dense(4, activation='softmax'))
model.summary()

model.compile(optimizer='rmsprop', loss='categorical_crossentropy',
metrics=['accuracy'])
model.fit(x_train, y_train, batch_size=5, epochs=20, validation_data=(x_test, y_test))
model.save("brain_tumor_dec_model.h5")
```

```
y_test_results = model.predict(x_test)

c = 10
fig, subplots = plt.subplots(1, c)
fig.set_size_inches(30, 9)
for i in range(c):
    n = np.random.randint(0, len(x_test))
    guess = np.argmax(y_test_results[n])
    actual = np.argmax(y_test[n])

    subplot = subplots[i]
    subplot.imshow(x_test[n].reshape((resolution, resolution)), cmap="gray")
    subplot.set_title(f"predicted: {guess}, actual: {actual}")
    subplot.axis("off")
plt.show()
```