



*Project Report On*

## **Multi-Model Cancer Detection**

*Submitted in partial fulfillment of the requirements for the  
award of the degree of*

**Bachelor of Technology**

*in*

***Computer Science and Engineering***

**By**

**Ajith Varghese Abraham (U2103018)**

**Anna Prince (U2103040)**

**Arjun Martin (U2103047)**

**Athulya S Pai (U2103055)**

**Under the guidance of**

**Mr. Harikrishnan M**

**Computer Science and Engineering**

**Rajagiri School of Engineering & Technology (Autonomous)**  
(Parent University: APJ Abdul Kalam Technological University)

**Rajagiri Valley, Kakkanad, Kochi, 682039**

**April 2025**

# CERTIFICATE

*This is to certify that the project report entitled "**Multi-Model Cancer Detection**" is a bonafide record of the work done by **Ajith Varghese Abraham (U2103018)**, **Anna Prince (U2103040)**, **Arjun Martin (U2103047)**, **Athulya S Pai (U2103055)**, submitted to the Rajagiri School of Engineering & Technology (RSET) (Autonomous) in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology (B. Tech.) in "**Computer Science and Engineering**" during the academic year 2024-2025.*

**Project Guide**

Mr. Harikrishnan M  
Assistant Professor  
Dept. of CSE  
RSET

**Project Co-ordinator**

Mr. Harikrishnan M  
Assistant Professor  
Dept. of CSE  
RSET

**Head of the Department**

Dr. Preetha K G  
Professor  
Dept. of CSE  
RSET

## **ACKNOWLEDGMENT**

We wish to express our sincere gratitude towards **Rev. Dr. Jaison Paul Mulerikkal CMI**, Principal of RSET, and **Dr. Preetha K G**, Head of the Department of Department of Computer Science and Engineering for providing us with the opportunity to undertake our project, "Multi-Model Cancer Detection".

We are highly indebted to our project coordinator, **Mr. Harikrishnan M**, Assistant Professor, Department of Computer Science and Engineering, for his valuable support.

It is indeed our pleasure and a moment of satisfaction for us to express our sincere gratitude to our project guide **Mr. Harikrishnan M** for his patience and all the priceless advice and wisdom he has shared with us. We also express our sincere thanks to **Ms. Meenu Mathew** Assistant Professor, Department of Computer Science and Engineering, for her support.

Last but not the least, we would like to express our sincere gratitude towards all other teachers and friends for their continuous support and constructive ideas.

**Ajith Varghese Abraham**

**Anna Prince**

**Arjun Martin**

**Athulya S Pai**

## Abstract

The project "Multi-Model Cancer Detection" aims to use advanced deep learning techniques to improve early detection of brain and lung cancers using medical imaging data, specifically MRI and CT scans. Current diagnostic methods are often time-consuming and prone to errors, creating a pressing need for automated and reliable systems. This project employs multiple state-of-the-art Convolutional Neural Network (CNN) architectures, including ResNet50, VGG16, Inception v3, and YOLO v8, to detect cancerous patterns efficiently and accurately. By comparing the performance of these models, the project identifies the most effective architecture for multi-organ cancer detection.

The methodology includes training and fine-tuning pre-trained models on standardized medical image datasets while addressing challenges such as data collection, preprocessing inconsistencies, and high computational demands. Preliminary results aim to highlight the strengths of each model in terms of accuracy, precision, and recall for detecting specific cancers. Ultimately, the project aspires to contribute to healthcare by providing a robust system capable of supporting clinicians with faster and more precise cancer diagnostics.

# Contents

<b>Acknowledgment</b>	<b>i</b>
<b>Abstract</b>	<b>ii</b>
<b>List of Abbreviations</b>	<b>vi</b>
<b>List of Figures</b>	<b>vii</b>
<b>List of Tables</b>	<b>viii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Background . . . . .	1
1.2 Problem Definition . . . . .	2
1.3 Scope and Motivation . . . . .	2
1.4 Objectives . . . . .	3
1.5 Challenges . . . . .	4
1.6 Assumptions . . . . .	4
1.7 Societal / Industrial Relevance . . . . .	4
1.8 Organization of the Report . . . . .	5
<b>2 Literature Survey</b>	<b>6</b>
2.1 Inception - V3 . . . . .	6
2.1.1 Introduction: . . . . .	6
2.1.2 Methodology: . . . . .	7
2.1.3 Challenges: . . . . .	8
2.1.4 Advantages: . . . . .	8
2.1.5 Results: . . . . .	8
2.1.6 Conclusion: . . . . .	9
2.2 VGG-16 . . . . .	9

2.2.1	Introduction . . . . .	9
2.2.2	Methodology . . . . .	9
2.2.3	Challenges . . . . .	10
2.2.4	Advantages . . . . .	10
2.2.5	Results . . . . .	10
2.2.6	Conclusion . . . . .	11
2.3	ResNet-50 . . . . .	11
2.3.1	Introduction . . . . .	11
2.3.2	Methodology . . . . .	11
2.3.3	Architecture: . . . . .	12
2.3.4	Pooling and Classification: . . . . .	12
2.3.5	Challenges . . . . .	12
2.3.6	Advantages . . . . .	12
2.3.7	Results . . . . .	12
2.3.8	Conclusion . . . . .	12
2.4	YOLO-V8 . . . . .	13
2.4.1	Introduction . . . . .	13
2.4.2	Methodology . . . . .	13
2.4.3	Challenges . . . . .	14
2.4.4	Advantages . . . . .	14
2.4.5	Results . . . . .	14
2.4.6	Conclusion . . . . .	14
2.5	Comparitive Study . . . . .	14
2.5.1	Introduction . . . . .	14
2.5.2	Findings . . . . .	15
2.5.3	Challenges . . . . .	15
2.5.4	Conclusion . . . . .	15
2.5.5	Overall Results . . . . .	15
2.6	Summary and Gaps Identified . . . . .	15
2.6.1	Conclusion . . . . .	16
2.6.2	Gaps Identified . . . . .	16

<b>3 System Design</b>	<b>21</b>
3.1 Introduction . . . . .	21
3.2 System Architecture . . . . .	22
3.3 Component Design . . . . .	24
3.4 Algorithms . . . . .	27
3.4.1 Algorithm for ResNet-50 . . . . .	27
3.4.2 Algorithm for Inception-V3 . . . . .	30
3.4.3 Algorithm for VGG-16: . . . . .	33
3.4.4 Algorithm for YOLOv8: . . . . .	35
3.5 Sequence Diagram . . . . .	39
3.6 Tools and Technologies . . . . .	39
3.7 Data Set . . . . .	40
3.8 Module Division . . . . .	41
3.9 Key Deliverables . . . . .	41
3.10 Gantt Chart . . . . .	42
<b>4 Results and Discussions</b>	<b>43</b>
4.1 Comparative Analysis . . . . .	43
4.2 System Outcomes . . . . .	44
<b>5 Conclusions &amp; Future Scope</b>	<b>46</b>
5.1 Conclusion . . . . .	46
5.2 Future Scope . . . . .	47
<b>6 Conclusion</b>	<b>48</b>
<b>References</b>	<b>48</b>
<b>Appendix A: Presentation</b>	<b>50</b>
<b>Appendix B: Vision, Mission, Programme Outcomes and Course Outcomes</b>	<b>77</b>
<b>Appendix C: CO-PO-PSO Mapping</b>	<b>81</b>

## List of Abbreviations

MRI	-	Magnetic Resonance Imaging
CNN	-	Convolutional Neural Network
DCNN	-	Deep Convolutional Neural Network
V3	-	Version 3
CT	-	Computed tomography
YOLOv8	-	You Only Look Once version 8
ResNet50	-	Residual Network-50
VGG-16	-	Visual Geometric Group-16

## List of Figures

2.1	Inception-V3	7
2.2	VGG-19	9
2.3	Resnet-50	11
2.4	YOLOv8	13
3.1	Architectural Diagram	22
3.2	Sequence Diagram	39
3.3	Gantt Chart	42
4.1	Comparison between the 4 models for Brain Cancer Detection	43
4.2	Comparison between the 4 models for Lung Cancer Detection	44

## **List of Tables**

2.1 Comparison of Deep Learning Techniques in Medical Imaging . . . . .	18
---	----

# **Chapter 1**

## **Introduction**

This project called "Multi-Model Cancer Diagnosis" seeks to detect cancer in the very early stages as well as an accurate diagnosis in vital organs like the brain and lungs, all with the efficacious use of advanced deep learning techniques. Its focus is on capturing the automated processing of diagnostic imaging modalities using MRI or CT scans for developing a system to eliminate things like inefficiency and mistake that generally come with the use of traditional diagnostic methods. The project focuses on several state-of-the-art convolutional neural network architectures, such as ResNet50, VGG16, Inception V3, and YOLOv8, to analyze and classify medical images for cancer detection.

This project compares different models to find the best architecture for multimodal cancer detection in organs. After evaluation on several performance parameters such as sensitivity, specificity, and accuracy, it will compare the capability of the model tumor identification and classification performances. This study is aimed at impacting heavy problems such as data variability, computational complexity, and compliance to norm, thus making it very robust and scalable while providing a solution to cancer diagnosis. The outcome of an effective early detection is improvement in treatment outcomes which later translates to improving the survival chances of the patients.

### **1.1 Background**

The project focused on development of an automation system to be used in deep learning approaches towards detection of both brain cancer and lung cancer. The models used are those on the MRI and CT scans, trained and analyzed using ResNet50, VGG16, Inception V3, and YOLO V8. Such models would acquire those cancerous patterns with much higher accuracy and efficiency. Therefore, the project is stepping forward to compare the performance of these different models in order to find out which architecture will perform

the best in diagnosing the different cancer types. One of the key objectives is to help achieve early diagnosis, which improves the efficacy of treatment, but this is an attempt towards preventing inefficiencies and errors that are part of the traditional method of diagnosis.

## 1.2 Problem Definition

- Challenge : Being time-lagged, there is inaccuracy and inefficiency in the present methods of diagnosis that lead to hindrance in early diagnosis of brain and lung cancers which, as a result, pose serious problems for the patients.
- Objective: To develop an automated system through deep learning whereby it will diagnose and identify the cancer tissues that have invaded the brain and lungs using MRI scans and CT scans as the sources of images.
- Technical Goal: Performance evaluation of the different deep learning architectures such as ResNet50, VGG16, Inception v3 and YOLOv8, concerning multi-organ cancer diagnosis for identification of the model most suitable for the task.
- Constraints: Resistances towards issues like unavailability of data, pre-processing needs, computation complexity, and regulations.
- Outcome: Accurate performing model for early-stage cancers so as to facilitate decisions for patient management.

## 1.3 Scope and Motivation

### SCOPE

- Goal: An approach that will automate diagnosis and detection using MRI and CT scans for brain and lung cancers.
- Methodology: Adoption of deep learning approaches through using different models, namely, ResNet50, VGG16, Inception v3, and YOLOv8.
- Evaluation: Comparisons in performance between the generic models for carrying identification as to which of them can be effectively used for effective detection existing cancer in above organs.

- Output: Accurate prediction with different advantages relating to models and performance measures (accuracy, precision, recall) in cancer detection.
- Tools: Up-to-date frameworks for deep learning such as TensorFlow, Keras, and Py-Torch will be put to use in implementation.

## MOTIVATION

- Diagnosis at an initially early stage is one of the most important factors that increase an individual's chance of survival from brain and lung cancer.
- Conventional methods of diagnosis are time-consuming and prone to human error, thus resulting in inaccurate diagnoses.
- Deep learning can possibly fully automate and speed up the process towards accurate diagnosis but within shorter times.
- Hence, the research study begins to link itself in the development of AI in healthcare, indeed, addressing today's inefficiencies in human diagnosis.

### 1.4 Objectives

- Automation for cancer detection: Setting up an automated system for detecting and diagnosing brain and lung cancers on the basis of MRI scans and CT scans.
- Performance evaluation: Employ and evaluate deep learning models, namely ResNet50, VGG16, Inception v3 and YOLO V 8, for cancer detection.
- A comparative analysis to compare models in terms of accuracy and efficiency to establish the best model for multi-organ cancer detection.
- Data processing as preprocessing diagnostic images is conducive to easy loading into deep learning models while ensuring uniformity in imaging quality.
- Performance Metrics: Monitoring and fine-tuning based under various relevant metrics such as accuracy, precision, and recall.
- Generalization of Models: Ensuring generalization of models in diverse datasets for validation of robustness in multi-organ cancer detection.

## **1.5 Challenges**

- Aggregating data - Annotated high quality MRI and CT scans become more and more scarce.
- Data Cleaning: Varying image quality and resolution also add noise in the collected data.
- Computational Complexity: High resources required to train a deep learning model.
- Regulatory and Ethical Issues: Compliance with laws regarding privacy and medical standards is required.

## **1.6 Assumptions**

- There would be considerable availability of data in wide-ranging and sufficient collections for MRI and CT scans.
- Homogeneous Image Quality: The same level of image quality among the institutions.
- Effective Fine Tuning: Models have also been pre-trained before they are applied for adapting the cancer detection task.

## **1.7 Societal / Industrial Relevance**

- Diagnosis at an Early Stage: Once you diagnose a brain and lung cancer, you get treated more efficaciously and have a better chance at survival.
- Health Care-Efficiency: Automated systems can cut down the window of diagnosis errors, thereby saving time for medical professionals.
- Integration of Technology: To promote advanced AI techniques to be used in the health care industry pertaining to improving diagnostic practices.

## **1.8 Organization of the Report**

- Introduction: This sets the problem and gives significant purposes and objectives of the study.
- Literature Review: Draw out previous studies and methods done so far in the area of cancer detection.
- Methodology: This would include much discussion of different models pertaining to deep learning and the usage of ResNet50, VGG16, Inception V3, YOLOv8.
- System Architecture: articulate clear architecture with use-case diagrams.
- Modules: At least one topical discussion on the architecture regarding the specific models.
- Assumptions and Challenges: This would be mostly considerations and barriers faced.
- Expected Output: Expected outcomes including a model comparison along with accuracy metrics.
- Implementation Details: Work breakdown, time schedule (Gantt chart) as well as present level of progress.
- Results and Conclusion: Important insights from partial implementation and project summary.
- References: Research articles and investigations cited.

# **Chapter 2**

## **Literature Survey**

A multi-modal cancer detection project on the aforestated subject deals with the emergencies of detecting brain and lung cancers at an early stage and with accuracy through advanced deep learning techniques.” The diagnostic methods of the present day are a lot inefficient, consume a lot of time, and are prone to human errors, thereby liable to create delays in treatment and reduce survival rate. This study takes into account the convolutional neural network architectures such as ResNet50, VGG16, Inception v3, and YOLOv8 applied for diagnostic image analysis: MRI and CT scan images. The automated detection process is going to enhance accuracy and speed for cancer diagnosis across organs.

The objective of the project is to evaluate these models with an aim to arrive at the best architecture for detecting cancer in important organs. Apart from that there are issues regarding different kinds of very high-quality annotated data, because of the compute complexity and the medical standards these have to be tackled in this research. The work completed is at 30% completion and has established the framework for these models and generated comparisons based on performance metrics. The expected outcome is high accuracy and thus a thorough understanding of the strengths of each model regarding types of cancer thus causing a paradigm shift in the field of cancer diagnostics.

### **2.1 Inception - V3**

#### **2.1.1 Introduction:**

[1] Convolutional Neural Network, which is optimal towards feature extraction, is Inception V3. It used a modular way to increase depth and breadth, which is why it excels in both fine-grained and high-level features from an image-from-MRI scope.

Inception-v3 is among the best optimized convolutional neural networks that provide improved feature extraction without much computational cost. The application of the modular approach is meant to improve the network's depth and width, thus providing high performance for both fine-grained and coarse-level features extraction from images-from-MRI scopes.

### 2.1.2 Methodology:

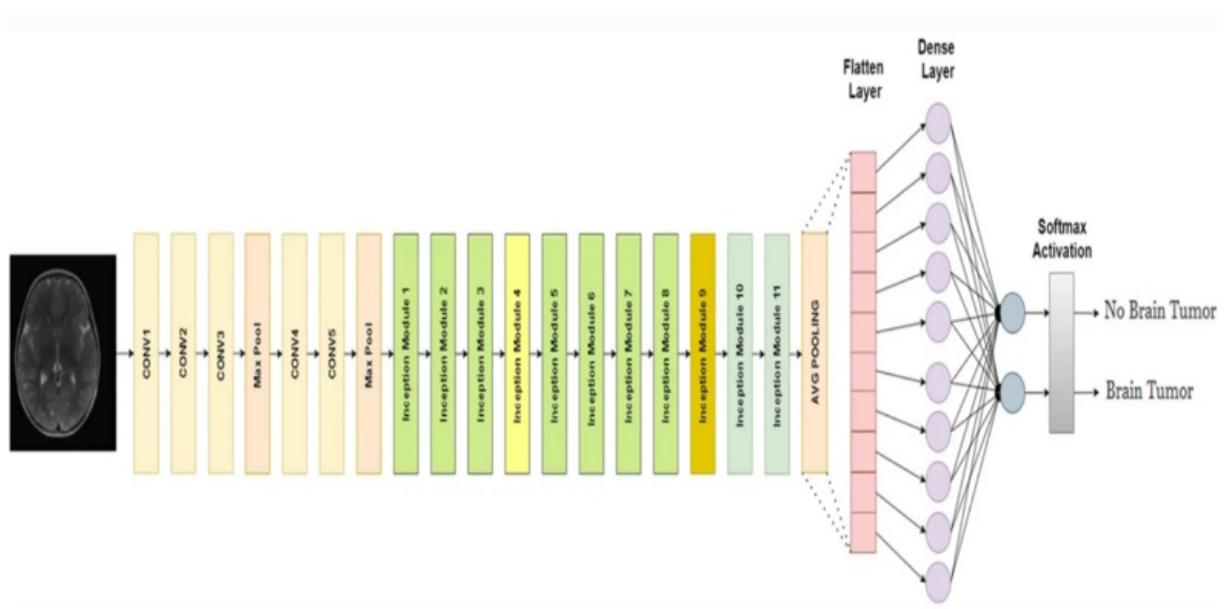


Figure 2.1: Inception-V3

### Convolutional Layers:

- Convolutions for feature extraction at multiple stages.
- In Step by Pooling: reducing the spatial dimensions step by step.

### Inception Modules:

- They use combined 1x1, 3x3, and 5x5 kernels for siphoning.
- Saves in number-the larger operations replaced by smaller ones continue to preserve information.

### **Auxiliary Classifiers:**

- Acts as those auxiliary additions to stop vanishing gradient during some intermediate stage.

### **Final Layers:**

- A fully connected layer followed by a softmax classifier to offer predictions.

#### **2.1.3 Challenges:**

- Heavy computational loads can be observed due to very deep architecture.
- Requires a large annotation dataset to optimally attain performance.
- Summarized optimization of the inception module's parameters is certainly significant.

#### **2.1.4 Advantages:**

- One of the advantages of the modular architecture of the system is that it simplifies learning and scalability.
- Enhancements in parallel convolution operations without compromising the accuracy of the task.
- Very good for macro and micro-feature extraction in medical imaging.

#### **2.1.5 Results:**

Inception v3 showed greater proficiency in classifying as well as detecting brain tumors which further demonstrate its potential to differentiate between healthy and cancerous tissues.

### 2.1.6 Conclusion:

This model is purposely created to manage greater image diagnostic work with optimal imbalance in performance and computational demand.

## 2.2 VGG-16

### 2.2.1 Introduction

VGG-16 [2] has finally become a historic model of CNN. Everything is going to depth and small-sized convolutional filters used for feature detection purposes.

### 2.2.2 Methodology

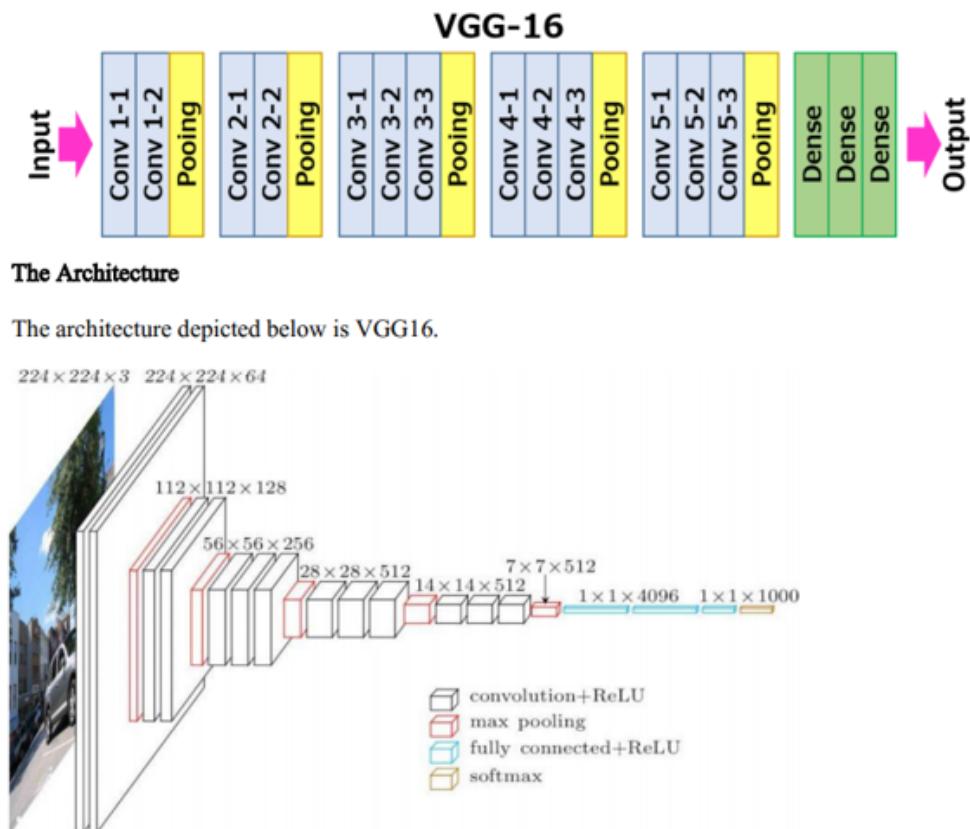


Figure 2.2: VGG-19

### **Layer Structure:**

It consists of 13 convolutional layers with all tiny filters measuring only 3 by 3 pixels in each dimension. The architecture finally terminates with three fully connected layers that separate the representations for the ensuing classification.

### **Pooling:**

Downsample feature maps using 2x2 max pooling layers after every few convolution layers.

### **Activation:**

As non-linearity is introduced by ReLU, this makes the network able to learn highly complex patterns.

### **Classification:**

After this last phase made of fully connected layers, we automate to the softmax output layer for classification purposes.

#### **2.2.3 Challenges**

- Requirement of very high memory due to a very large number of parameters.
- Time required to train VGG-16 is much longer when compared to modern architectures, e.g., ResNet.

#### **2.2.4 Advantages**

- Quite simple implement and adapt to specific tasks.
- It has the ability to pick small details for intelligent performance in medical imaging.

#### **2.2.5 Results**

- Results of analysis showed that VGG-16 is very well exhibiting precision and recall behavior in the detection of brain cancer. However, it requires an exhaustive amount of training and otherwise resource-demanding inference on VGG-16.

## 2.2.6 Conclusion

VGG-16 is still in the race for fairly important image classification tasks, especially in a domain such as medical diagnosis needing tight granularity. But it comes at a heavy computational cost.

## 2.3 ResNet-50

### 2.3.1 Introduction

ResNet-50[3] has taken the deep learning paradigm to a different level with the advent of residual learning that procures training of very deep networks without the misery of the vanishing gradient problem, making it useful for several complex medical imaging classification tasks.

### 2.3.2 Methodology

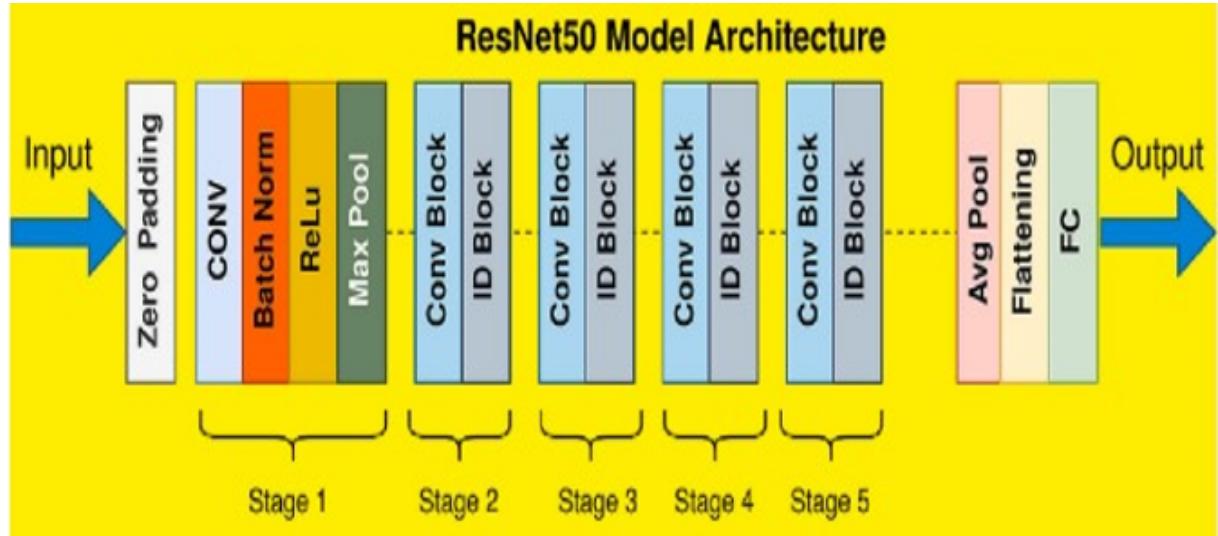


Figure 2.3: Resnet-50

### Residual Learning:

- Skip connections provide an undistorted path for gradients passing through the network, bypassing layers.
- Can be effectively applied to training networks of very large depth.

### **2.3.3    Architecture:**

- Five stages amalgamate convolution, batch normalization, and ReLU layers.
- Residual blocks augmented by identity shortcuts reduce the complexity in learning features.

### **2.3.4    Pooling and Classification:**

- Global average pooling reduces the data to an extremely low dimension.
- The last classification prediction is provided by a fully connected layer.

### **2.3.5    Challenges**

- Needs heavy computational power due to its depth.
- It tends to be overfitted if it is not properly regularized or trained through a sufficiently large set.

### **2.3.6    Advantages**

- It trains models at a great depth but does not degrade performance and results in state-of-the-art accuracy of the task.

### **2.3.7    Results**

In distinguishing and classifying from MRI images, the ResNet-50 model performed well as normal or positive for brain tumors, hence showing very high accuracy measures along with their recall rates. It also performed well in distinguishing healthy tissue against cancerous tissues.

### **2.3.8    Conclusion**

ResNet-50 is an another very great resource under medical diagnostics using residual learning since it can perform quite difficult tasks such as cancer detection reliably.

## 2.4 YOLO-V8

### 2.4.1 Introduction

YOLO-v8[4] is a new framework for object detection that is intended for real-time applications. It is designed for high speed and accuracy. This performance is enough to develop a tumor detection system from medical images.

### 2.4.2 Methodology

#### Architecture:

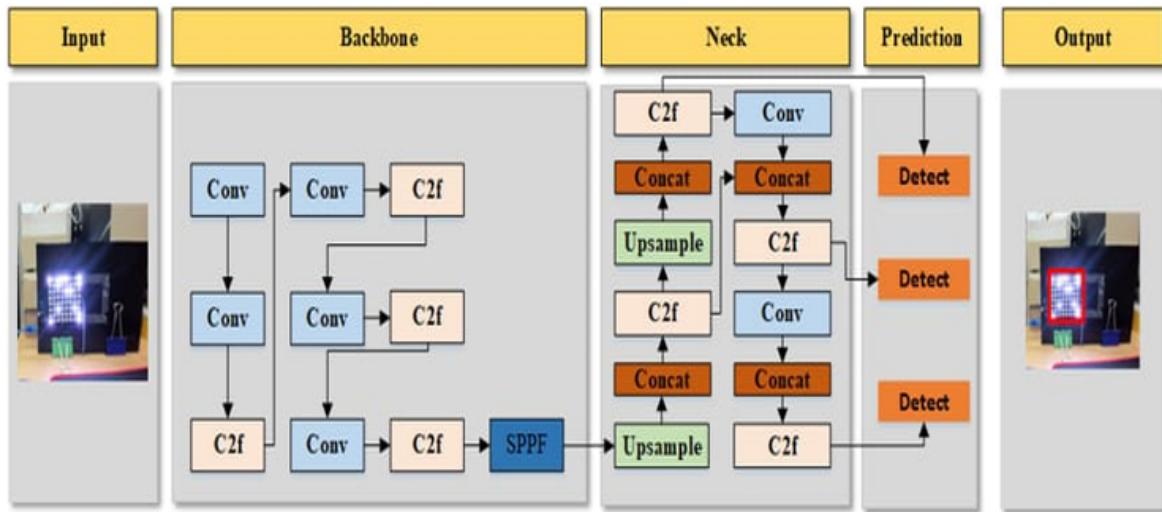


Figure 2.4: YOLOv8

#### Backbone:

Extensions of low, medium and high-level feature maps through pre-trained CNNs.

#### Neck:

Combining feature maps through methods like Feature Pyramid Network (FPN).

### **Head:**

Classification and bounding box prediction of an object.

### **Training:**

- Annotated datasets for training the detection head.
- Improving classification and localization accuracies

#### **2.4.3 Challenges**

- Data quality is needed for high-grade annotated datasets.
- It becomes problematic to detect very small or overlapped objects.

#### **2.4.4 Advantages**

- Real-time with high accuracy detection.
- Flexibility and scalability regarding various image sizes and shapes of objects.

#### **2.4.5 Results**

In the cancer identification and localization tasks for both MRI and CT scans acquired real-time diagnostics with accurate bounding boxes that demonstrated very good performance.

#### **2.4.6 Conclusion**

Examining efficiency and accuracy, YOLO-v8 becomes an efficient tool for real-time tumor detection via medical imaging.

### **2.5 Comparative Study**

#### **2.5.1 Introduction**

This study[5] is undoubtedly between Inception-v3, VGG-16, ResNet-50, and YOLO-v8 in multi-organ cancer detection in terms of the and weaknesses in identifying brain and lung cancer from diagnostic images.

### **2.5.2 Findings**

#### **Inception-v3:**

Balanced performance in extracting the features effectively.

#### **VGG-16:**

A lot of precision but very close to impossible to compute.

#### **ResNet-50:**

Owes well classifying using deep architecture.

#### **YOLO-v8:**

Best for real time detection and very precise bounding box prediction.

### **2.5.3 Challenges**

These models will need to be tuned and pre-processed to a very large extent since they all differ in their variability on the medical imaging datasets.

### **2.5.4 Conclusion**

Thus, the study depicts YOLO-v8 and ResNet-50 as the two most effective models for detecting cancer in multi-organ settings; each model focusing on differentiated detection versus classification strengths.

### **2.5.5 Overall Results**

It is indicated that hybridization that will account for the strength of both models will serve in producing the best automated cancer detection systems.

## **2.6 Summary and Gaps Identified**

### **2.6.1 Conclusion**

Using in-depth analyzing techniques like MRI and CT scans, the Multi Modal Cancer Detection project aims to help detect and diagnose cancers in two vital organs—the brain and lungs. By putting up a complete comparison and evaluation of different state-of-the-art architectures, including Inception-v3, YOLO-v8, ResNet-50, and VGG-16. The study identifies the pressing need for accurate, swift diagnostic tools and established the merits of each model; for instance real-time detection capabilities provided by YOLO-v8, rugged classification accuracy exhibited by ResNet-50, etc. It has been observed that generally no one model is found to perform better than other models; instead, the choice of the model rests on specific task requirements, either it should be high in precision while classifying or speedy in cases of real-time detection. It indicates the feasibility of putting together several design frameworks to advance a hybrid architecture with extremely effective and dependable accuracy.

### **2.6.2 Gaps Identified**

#### **Limited Dataset Diversity:**

Primarily, this project targets a database of images taken from MRI and CT scan. Yet, it is likely that there will be a deficiency in the dataset whereby diversity in scanner types, images, and various demographics for a few patients would write many biases into the performance of the model.

#### **Dependence on Pre-trained Models:**

Dead Wood on a Reliability of Pre-Trained Models Pre-trained models, i.e., ResNet-50 VGG-16 and YOLO-v8, may fast-track this process, unlike making time-consuming adjustments with their own imaging knowledge or goodness of fit in respect of detection and generalizability.

#### **Computational Resource Demands:**

Training and fine-tuning of deep-learning models literally suck in computational resource. Thus, the exercise poses great scalability challenges and limits accessibility for less resourceful institutions.

### **Validation in Real World:**

Even though models are being tested on some datasets, looking at the amounts tested through multiple clinics, the real world validity of such models throws up a big question mark on their usability and applicability.

### **Ethical and Privacy Issues:**

There are many laws and regulations, such as HIPAA or GDPR, regarding privacy and security of data that give challenges while handling sensitive medical data.

### **Challenge of Small Tumor Detection:**

Models such as YOLO-v8 may have difficulty in detecting and localizing tumors that are very small or overlapping, although they are important for the primary diagnosis for cancer cases.

### **Overfitting Risks:**

In certain cases, insufficient amounts of labelled datasets pertaining to certain types of cancers may also not be enough and hence increase the chances for overfitting diminishing the chances of generalization to new data.

Table 2.1: Comparison of Deep Learning Techniques in Medical Imaging

Technique	Advantages	Disadvantages
Inception-v3	<ul style="list-style-type: none"> <li>• High modular architecture to efficiently extract features.</li> <li>• Harmony in depth and breadth for analysis at a finer granularity.</li> <li>• Scale well for high-resolution datasets .</li> </ul>	<ul style="list-style-type: none"> <li>• Complicated computationally demanding lots of resources.</li> <li>• Complex hyperparameter optimization processes.</li> </ul>
YOLO-v8	<ul style="list-style-type: none"> <li>• Tumor detection in real time with high speed and accuracy.</li> <li>• Flexible and scalable for diverse medical imaging datasets.</li> <li>• Precise bounding box predictions for object localization.</li> </ul>	<ul style="list-style-type: none"> <li>• It is heavily dependent on the high-quality annotated datasets.</li> <li>• It faces problems in detecting overlapping or small objects.</li> </ul>

Technique	Advantages	Disadvantages
ResNet-50	<ul style="list-style-type: none"> <li>• Stable training with deep residual learning.</li> <li>• Very accurate in classifying tasks.</li> <li>• Very inspiring for the vanishing gradient problems.</li> </ul>	<ul style="list-style-type: none"> <li>• Needs a lot of computational power.</li> <li>• Prone to overfitting on small datasets without regularization.</li> </ul>
VGG-16	<ul style="list-style-type: none"> <li>• Very easy and uniform architecture which can be easily adapted.</li> <li>• Quite effective in fine-grained details.</li> <li>• Dependable task in medical diagnostic.</li> </ul>	<ul style="list-style-type: none"> <li>• High memory and computational requirements</li> <li>• Slower training process compared to modern architectures.</li> </ul>

<b>Technique</b>	<b>Advantages</b>	<b>Disadvantages</b>
Comparative Study	<ul style="list-style-type: none"> <li>• Finds the organ-specific strengths of the model in multi-organ cancer detection.</li> <li>• Exploring the tradeoffs in terms of accuracy and efficiency.</li> <li>• Possibility of hybrid approaches.</li> </ul>	<ul style="list-style-type: none"> <li>• Comparison analysis depends on such availability of standardized datasets.</li> <li>• Requires further improvement of real-world systems.</li> </ul>

# **Chapter 3**

## **System Design**

### **3.1 Introduction**

Now, diagnosis and treatment revolve around medical imaging; brain tumors and lung cancer are two major diseases in this category because of their impact on health, and early detection ensures great outcomes. Today, advanced machine learning (ML) systems have proved their magic in the analysis of medical images by identifying and classifying abnormalities at a remarkable level of precision. The systems also have automated, efficient, and accurate analysis for MRI and CT scans using a combination of techniques that include convolutional neural networks (CNNs), and pretrained models such as ResNet50, InceptionV3, and YOLOv8.

It is a general workflow involving the acquisition of data, preprocessing, features extraction, and then model training in order to use efficient classifiers and effective performance evaluation metrics to be able to comprise a complete system ideally meant to supplement and guide the medical practitioners towards the timely and accurate diagnosis. This is a fine application of artificial intelligence medicine that will be beneficial in improving patients' care as well as developing health systems in reality around the world.

### 3.2 System Architecture

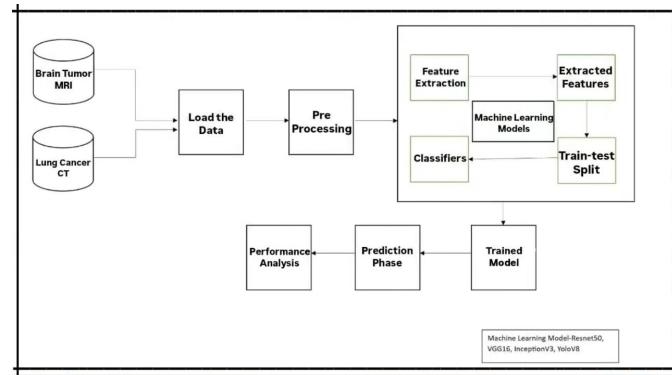


Figure 3.1: Architectural Diagram

The diagram serves to illuminate the architecture for regular workflow in a machine learning (ML) system:

Detecting and classifying brain tumors using MRI images and lung cancer using CT images.

#### 1. Data Sources:

Brain Tumor (MRI): This block acts as a source for MRI images of the brain which have been suspected to contain tumors.

Lung Cancer (CT): This block acts as a source for CT images of lungs containing either potentially cancerous nodules or not.

#### 2. Load the data:

This step involves directly loading the primary raw image data, but rather than targeting specific portions to provide individualized views of the targeted sources, it has image file reading and storage in the preferred storing format for further processing stages ( JPG ).

#### 3. Preprocessing:

Image Preprocessing: This is the most critical stage of processing the raw images for the input into the ML model. Basic Points include:

Rescaling: Altering the image dimensions to the exact same size in which they will be fed into the model.

Normalization: Truncating the pixel intensity patterns to a specific dimension or range, for example from 0 to 1, so that it can be fed into the model to augment its training.

Augmentation: This is the creation of variations of the original image itself:

eg: rotations, flips and so on, as this leads to size increase in the dataset and improves model robustness.

Noise Reduction: Removal of any unwanted image noise.

#### **4. Feature extraction:**

This block holds the extraction of the required features of the preprocessed images; these features would be numerical representations of the main characteristics captured in an image, such as shapes, textures, and intensity patterns. Some of these methods include the following: Handcrafted Features: Features are extracted based on specific applications-good examples are gray-level co-occurrence matrices (GLCM) or Local Binary Patterns (LBP).

#### **5. Machine Learning Models:**

The essential mechanized system is this block where the features are gathered to model a machine learning system. The choice of model depends on the application and associated characteristics of the dataset. The commonly used models include:

Support Vector Machines (SVM): for classification. Random Forest: works for classification and regression both. Convolutional Neural Networks: pattern recognition for images, such as detection and segmentation. Recurrent Neural Networks: for sequential data, like time series.

#### **6. Classifiers:**

Medical Image Processors classify images according to classes like in the brain tumor cases; it has got two classes, 'tumor' and 'no tumor'. For instance, in lung cancer probing, classes can be noun-ed as 'cancerous nodule', 'benign nodule', 'no nodule'.

## **7. Split the training and test data:**

The two datasets are ideally created before any training of the model:

Training set: here, the training set trains the model and the patterns of the underlying data are learned for testing purposes.

Testing set: Used for evaluation of the performance of the trained model on previously unseen data.

## **8. Trained Model:**

The trained model is then saved to use in future analysis once the model has been trained against training data.

## **9. Prediction Phase:**

This phase is all about giving an input image and predicting new unseen images from that input image using the classification rules learned from features that have been extracted.

## **10. Performance Analysis:**

The next part is its predicted output. This output is tested on test datasets and is measured for performance metrics such as accuracy, precision, recall, and F1-score. As such a standard measure sets how much a model diagnoses a disease from the level of false positives and false negatives to confirm the model very well.

## **11. Manifested Features:**

This is the output of feature extracting methods. The features extracted are fed into the machine learning model for training and predicting.

### **3.3 Component Design**

#### **1. Sources of Data:**

Brain Tumor (MRI): This is the representation of the MRI images carrying the brain tumors.

Lung Cancer (CT): These are the collections of some CT scans with possible lung cancer.

## **2.Load data:**

Loading and reading the medical images from their respective storage locations. An image is normally in JPG (joint photographic group) format.

## **3.Preprocessing:**

Image Enhancement: It is a processing method by which alteration occurs in the brightness and noise levels in an image, and sharpness is added so that the image becomes good and abnormal structures can be visible.

Image Registration: Transfer all images to a common point of reference if more than one image is used (for example, different time points must be used).

Image Segmentation: The most critical step is to segment the part of interest i.e. brain. So, main possible regions for the segmentation of MRI scans are either brain or lungs from the background. This may be done using the thresholding, region growing, or more sophisticated active contours or deep learning-based segmentation techniques.

Data augmentation: Use among different image augmentation techniques, such as rotating, flipping, and zooming in on or out, with intensity shifting to augment images with different transformations, thus increasing the diversity of training data to improve the generalization of the model.

## **4. Feature Extractions:**

Manual Feature Engineering: Extraction of certain features from preprocessed images that are assumed to be relevant to the classification task, examples being texture features (gray-level co-occurrence matrix), shape features (area and perimeter) and statistical features (mean and standard deviation).

Deep Learning-Based Feature Extraction:It includes ResNet50, VGG16, InceptionV3, and YOLOv8. The deep learning models especially CNNs are good at learning, from raw images, the automatic hierarchical features.

## **5. Machine Learning Model:**

Classifier: The extracted features are created as a machine learning model to classify images as diseased or non-diseased.

Common classifiers are:

- Logistic Regression
- Deep Neural Networks
- Softmax
- ReLU

Training and Testing Split: Training and test parting of the data utilized where the model is trained on the training data and evaluated on the unseen testing one.

## **6. Train-Test Split:**

The labeled dataset is divided into two categories:

- Training Set - data to train the machine learning model; against blinded performance.
- Testing Set - How well trained algorithm predicts previously unseen data.

## **7. Trained Model:**

Once you have finished the training, use that model to make predictions on new, unseen images.

## **8. Prediction Phase:**

In this phase, the already trained model is used to classify unknown images as images containing the disease and images not having the disease.

## **9. Performance Analysis:**

The performance metrics on which this model will be evaluated include: Accuracy level metric-The percentage of correct classification analysis of the given images. Precision-The percent of actual positive examples to all predicted positive examples. Recall-The percent of true positives to all actual positives. F1-score-The Harmonic mean among precision and recall. ROC curve and the AUC Visual Representation of Capability Model Class Discrimination.

Model Tuning: Beyond evaluation on performance, hyperparameters can be tuned further to improve the accuracy of the model.

## 3.4 Algorithms

### 3.4.1 Algorithm for ResNet-50

#### Step 1: Input Preprocessing

```
\begin{itemize}
    \item INPUT: Image (H x W x C)
    \item RESIZE Image to (224 x 224 x 3)
    \item NORMALIZE pixel values to range [0, 1]
    \item STANDARDIZE using mean and standard deviation of the dataset
        \hookrightarrow (e.g., ImageNet)
\end{itemize}
%\vspace{0.5cm}
```

#### Step 2: Residual Block

```
# Implements a residual block with identity or projection shortcut
def residual_block(input, filters, stride=1, projection_shortcut=False):
    if projection_shortcut:
        shortcut = Conv2D(input, filters=filters,
                           kernel_size=(1, 1), stride=stride)
        shortcut = BatchNorm(shortcut)
    else:
        shortcut = input

    # First layer in the residual block
    conv1 = Conv2D(input, filters=filters,
                   kernel_size=(1, 1), stride=stride)
    conv1 = BatchNorm(conv1)
    conv1 = ReLU(conv1)

    # Second layer (3x3 convolution)
    conv2 = Conv2D(conv1, filters=filters,
```

```

        kernel_size=(3, 3), stride=1, padding='same')

conv2 = BatchNorm(conv2)
conv2 = ReLU(conv2)

# Third layer (1x1 convolution)
conv3 = Conv2D(conv2, filters=4*filters,
               kernel_size=(1, 1), stride=1)
conv3 = BatchNorm(conv3)

# Add shortcut to the output
output = Add(conv3, shortcut)
output = ReLU(output)
return output

```

### Step 3: Building ResNet-50

```

def resnet50(input, num_classes):
    # Initial Convolutional Layer
    conv1 = Conv2D(input, filters=64, kernel_size=(7, 7),
                   stride=2, padding='same')
    conv1 = BatchNorm(conv1)
    conv1 = ReLU(conv1)
    pool1 = MaxPool2D(conv1, pool_size=(3, 3),
                      stride=2, padding='same')

    # Stage 1
    stage1 = residual_block(pool1, filters=64,
                           stride=1, projection_shortcut=True)
    for _ in range(2): # Add 2 more blocks with identity shortcut
        stage1 = residual_block(stage1, filters=64)

    # Stage 2

```

```

stage2 = residual_block(stage1, filters=128,
                       stride=2, projection_shortcut=True)

for _ in range(3): # Add 3 more blocks with identity shortcut
    stage2 = residual_block(stage2, filters=128)

# Stage 3

stage3 = residual_block(stage2, filters=256,
                       stride=2, projection_shortcut=True)

for _ in range(5): # Add 5 more blocks with identity shortcut
    stage3 = residual_block(stage3, filters=256)

# Stage 4

stage4 = residual_block(stage3, filters=512,
                       stride=2, projection_shortcut=True)

for _ in range(2): # Add 2 more blocks with identity shortcut
    stage4 = residual_block(stage4, filters=512)

# Global Average Pooling and Fully Connected Layer

gap = GlobalAveragePooling2D(stage4)
output = Dense(gap, units=num_classes, activation=Softmax)

return output

```

#### Step 4: Training and Inference

```

\begin{itemize}
\item TRAIN model with categorical cross-entropy loss and optimizer
    \hookrightarrow (e.g., SGD with momentum or Adam)
\item INFER class probabilities for new inputs using trained model
\end{itemize}

```

### 3.4.2 Algorithm for Inception-V3

#### Step 1: Input Preprocessing

```
\begin{itemize}
    \item INPUT: Image (H x W x C)
    \item RESIZE Image to (299 x 299 x 3)
    \item NORMALIZE pixel values to range [-1, 1]
\end{itemize}
```

#### Step 2: Base Convolutional Layers

```
def base_convolution(input):
    conv1 = Conv2D(input, filters=32, kernel_size=(3, 3),
                   stride=2, activation=ReLU)
    conv2 = Conv2D(conv1, filters=32, kernel_size=(3, 3),
                   activation=ReLU)
    conv3 = Conv2D(conv2, filters=64, kernel_size=(3, 3),
                   activation=ReLU, padding='same')
    pool = MaxPool2D(conv3, pool_size=(3, 3), stride=2)
    return pool
```

#### Step 3: Factorized Convolutions

```
def factorized_convolution(input):
    fc1 = Conv2D(input, filters=80, kernel_size=(1, 1),
                  activation=ReLU)
    fc2 = Conv2D(fc1, filters=192, kernel_size=(3, 3),
                  activation=ReLU)
    pool = MaxPool2D(fc2, pool_size=(3, 3), stride=2)
    return pool
```

#### Step 4: Inception Module

```
def inception_module(input, filters_1x1, filters_3x3_reduce,
                     filters_3x3, filters_5x5_reduce,
                     filters_5x5, filters_pool_proj):
    path1 = Conv2D(input, filters=filters_1x1,
                   kernel_size=(1, 1), activation=ReLU)

    path2 = Conv2D(input, filters=filters_3x3_reduce,
                   kernel_size=(1, 1), activation=ReLU)
    path2 = Conv2D(path2, filters=filters_3x3,
                   kernel_size=(3, 3), activation=ReLU,
                   padding='same')

    path3 = Conv2D(input, filters=filters_5x5_reduce,
                   kernel_size=(1, 1), activation=ReLU)
    path3 = Conv2D(path3, filters=filters_5x5,
                   kernel_size=(5, 5), activation=ReLU,
                   padding='same')

    path4 = MaxPool2D(input, pool_size=(3, 3),
                      stride=1, padding='same')
    path4 = Conv2D(path4, filters=filters_pool_proj,
                   kernel_size=(1, 1), activation=ReLU)

    concat = Concatenate([path1, path2, path3, path4], axis='depth')
    return concat
```

#### Step 5: Auxiliary Classifier

```
def auxiliary_classifier(input, num_classes):
    pool = AveragePool2D(input, pool_size=(5, 5), stride=3)
    flat = Flatten(pool)
```

```

fc1 = Dense(flat, units=128, activation=ReLU)
dropout = Dropout(fc1, rate=0.7)
output = Dense(dropout, units=num_classes, activation=Softmax)
return output

```

### Step 6: Global Average Pooling and Output Layer

```

def final_classifier(input, num_classes):
    gap = GlobalAveragePooling2D(input)
    dropout = Dropout(gap, rate=0.4)
    output = Dense(dropout, units=num_classes, activation=Softmax)
    return output

```

### Step 7: Assembling the Model

```

def inception_v3_model(input, num_classes):
    base = base_convolution(input)
    factor_conv = factorized_convolution(base)

    inception1 = inception_module(factor_conv, 64, 48, 64,
                                   64, 96, 32)
    inception2 = inception_module(inception1, 128, 64, 128,
                                   64, 96, 64)
    pool = MaxPool2D(inception2, pool_size=(3, 3), stride=2)

    inception3 = inception_module(pool, 192, 96, 192,
                                   64, 96, 64)
    inception4 = inception_module(inception3, 192, 96, 192,
                                   64, 96, 64)

    aux_classifier = auxiliary_classifier(inception4, num_classes)
    final_classifier = final_classifier(inception4, num_classes)
    return final_classifier

```

## Step 8: Training and Inference

- TRAIN model with categorical cross-entropy loss and optimizer (e.g., Adam)
- INFER class probabilities for new inputs using trained model

### 3.4.3 Algorithm for VGG-16:

#### Step 1: Input Preprocessing

```
\begin{itemize}
    \item INPUT: Image (H x W x C)
    \item RESIZE Image to (224 x 224 x 3)
    \item NORMALIZE pixel values to range [0, 1]
    \item STANDARDIZE using mean and standard deviation of the dataset
        \hookrightarrow (e.g., ImageNet)
\end{itemize}
```

#### Step 2: Convolutional Block

```
function conv_block(input, num_filters, num_conv_layers):
    OUTPUT = input
    for i in range(num_conv_layers):
        OUTPUT = Conv2D(OUTPUT, filters=num_filters, kernel_size=(3, 3),
                       \hookrightarrow padding='same', activation=ReLU)
    OUTPUT = MaxPool2D(OUTPUT, pool_size=(2, 2), stride=2)
    return OUTPUT
```

#### Step 3: Fully Connected Layers

```
function fully_connected_block(input, num_classes):
    FLAT = Flatten(input)
    FC1 = Dense(FLAT, units=4096, activation=ReLU)
```

```

DROPOUT1 = Dropout(FC1, rate=0.5)
FC2 = Dense(DROPOUT1, units=4096, activation=ReLU)
DROPOUT2 = Dropout(FC2, rate=0.5)
OUTPUT = Dense(DROPOUT2, units=num_classes, activation=Softmax)
return OUTPUT

```

#### Step 4: Building VGG-16

```

function vgg16(input, num_classes):
    # Block 1: Two convolutional layers with 64 filters
    BLOCK1 = conv_block(input, num_filters=64, num_conv_layers=2)

    # Block 2: Two convolutional layers with 128 filters
    BLOCK2 = conv_block(BLOCK1, num_filters=128, num_conv_layers=2)

    # Block 3: Three convolutional layers with 256 filters
    BLOCK3 = conv_block(BLOCK2, num_filters=256, num_conv_layers=3)

    # Block 4: Three convolutional layers with 512 filters
    BLOCK4 = conv_block(BLOCK3, num_filters=512, num_conv_layers=3)

    # Block 5: Three convolutional layers with 512 filters
    BLOCK5 = conv_block(BLOCK4, num_filters=512, num_conv_layers=3)

    # Fully connected layers
    OUTPUT = fully_connected_block(BLOCK5, num_classes)

return OUTPUT

```

#### Step 5: Training and Inference

- TRAIN model with categorical cross-entropy loss and optimizer (e.g., SGD with momentum or Adam)

- INFER class probabilities for new inputs using trained model

### 3.4.4 Algorithm for YOLOv8:

#### Step 1: Input Preprocessing

```
\begin{itemize}
    \item INPUT: Image (H x W x C)
    \item RESIZE Image to target size (e.g., 640 x 640)
    \item NORMALIZE pixel values to range [0, 1]
\end{itemize}
```

#### Step 2: Backbone Network

```
function backbone(input):
    # Initial convolution
    CONV1 = Conv2D(input, filters=32, kernel_size=(3, 3), stride=2,
                   activation=ReLU)

    # CSP-based feature extraction
    CSP1 = csp_block(CONV1, filters=64, num_blocks=3)
    CSP2 = csp_block(CSP1, filters=128, num_blocks=6)
    CSP3 = csp_block(CSP2, filters=256, num_blocks=9)
    CSP4 = csp_block(CSP3, filters=512, num_blocks=3)

    return [CSP2, CSP3, CSP4] # Features at multiple scales

# CSP Block
function csp_block(input, filters, num_blocks):
    PART1 = Conv2D(input, filters=filters, kernel_size=(1, 1))
    PART2 = Conv2D(input, filters=filters, kernel_size=(1, 1))
```

```

for i in range(num_blocks):
    PART2 = residual_block(PART2, filters=filters)

CONCAT = Concatenate([PART1, PART2], axis='depth')
OUTPUT = Conv2D(CONCAT, filters=filters, kernel_size=(1, 1))
return OUTPUT

```

### Step 3: Neck (Feature Aggregation)

```

function neck(features):
    # Feature Pyramid
    P3, P4, P5 = features

    # Upsample and merge
    UP1 = Upsample(P5, scale=2)
    MERGE1 = Concatenate([UP1, P4], axis='depth')
    FPN1 = csp_block(MERGE1, filters=256, num_blocks=3)

    UP2 = Upsample(FPN1, scale=2)
    MERGE2 = Concatenate([UP2, P3], axis='depth')
    FPN2 = csp_block(MERGE2, filters=128, num_blocks=3)

    # Downsample and merge
    DOWN1 = Downsample(FPN2, scale=2)
    MERGE3 = Concatenate([DOWN1, FPN1], axis='depth')
    PAN1 = csp_block(MERGE3, filters=256, num_blocks=3)

    DOWN2 = Downsample(PAN1, scale=2)
    MERGE4 = Concatenate([DOWN2, P5], axis='depth')
    PAN2 = csp_block(MERGE4, filters=512, num_blocks=3)

return [FPN2, PAN1, PAN2] # Aggregated features at three scales

```

#### Step 4: Head (Detection and Classification)

```
function head(features, num_classes):  
    OUTPUTS = []  
    for feature in features:  
        # Predict bounding boxes, objectness scores, and class  
        # probabilities  
        PRED = Conv2D(feature, filters=(num_classes + 5) * num_anchors,  
                      kernel_size=(1, 1))  
        OUTPUTS.append(PRED)  
    return OUTPUTS
```

#### Step 5: Postprocessing

```
function postprocess(outputs, confidence_threshold, iou_threshold):  
    DETECTIONS = []  
    for output in outputs:  
        # Decode bounding box coordinates, objectness, and class  
        # probabilities  
        BOXES, SCORES, CLASSES = decode(output)  
        # Apply confidence threshold  
        FILTERED = filter_by_confidence(BOXES, SCORES, CLASSES,  
                                         confidence_threshold)  
        DETECTIONS.append(FILTERED)  
  
    # Apply Non-Maximum Suppression (NMS)  
    FINAL_DETECTIONS = apply_nms(DETECTIONS, iou_threshold)  
    return FINAL_DETECTIONS
```

#### Step 6: YOLOv8 Model

```
function yolov8(input, num_classes, confidence_threshold=0.25,  
                 iou_threshold=0.45):  
    BACKBONE_FEATURES = backbone(input)
```

```
NECK_FEATURES = neck(BACKBONE_FEATURES)

RAW_OUTPUTS = head(NECK_FEATURES, num_classes)

FINAL_OUTPUTS = postprocess(RAW_OUTPUTS, confidence_threshold,
    ↪ iou_threshold)

return FINAL_OUTPUTS
```

## Step 7: Training and Inference

```
\begin{itemize}
    \item During training: TRAIN model with a combination of loss
        ↪ functions:
            \begin{itemize}
                \item Localization loss (e.g., CIoU loss)
                \item Objectness loss (e.g., Binary Cross-Entropy)
                \item Classification loss (e.g., Cross-Entropy)
            \end{itemize}
    \end{itemize}
    \item During inference: INFER bounding boxes, objectness scores, and
        ↪ class probabilities for new inputs
\end{itemize}
```

### 3.5 Sequence Diagram

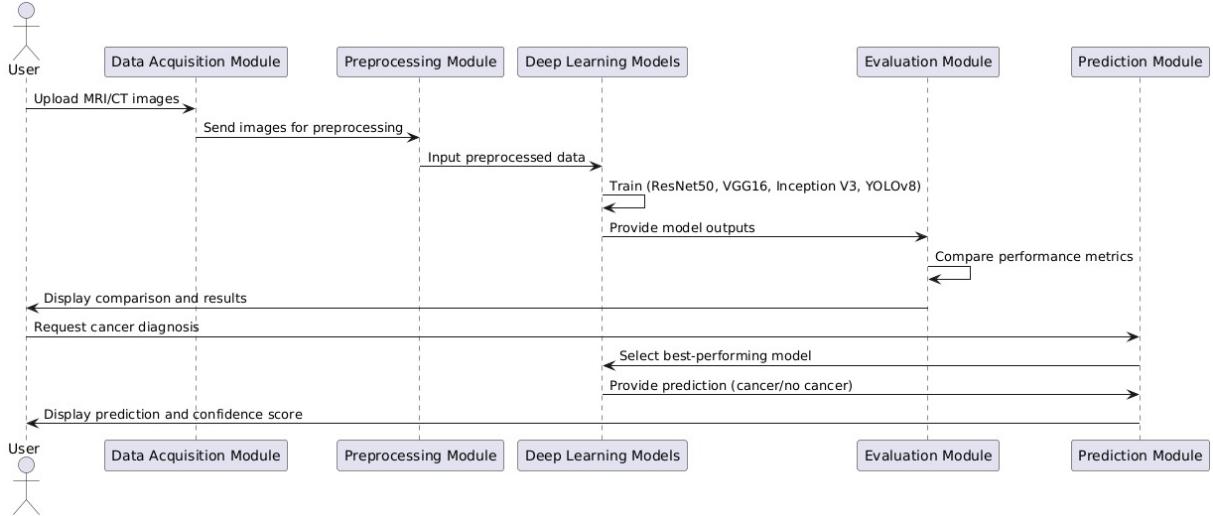


Figure 3.2: Sequence Diagram

### 3.6 Tools and Technologies

#### Hardware Requirements

CPU: A processor that is capable of having sufficient processing power is essential on the machine to handle processing from deep learning model training and deployment. It should qualify to include Intel Core i5 or equivalent AMD Ryzen 5 CPUs, at the very least. This guarantees that tasks such as feature extraction, model training, and inference run smoothly. Such a modern-day CPU should have multi-threading and very high-speed for usability.

GPU: A dedicated graphics processing unit otherwise called a GPU-accelerates and simplifies deep learning functions. The image depicts NVIDIA GTX 1060 or AMD Radeon RX 6800 XT-or higher specifications required. They speed up computations in parallel hence laying the foundation to compare training and inference speeds in hundreds to thousands of times faster than that of a computer itself.

RAM: It must have enough RAM to hold the model parameters, many intermediate results, and all the datasets used for training and inference. As 16 GB is the minimum recommendation because that would involve looking at how well most of the deep learning models and big datasets go, more would be better.

SSD: For storage speed, solid state drives are really much welcomed, SSD will enhance storage performance to the overall operation of a system. Size of SSD = Hoard amount = Size of models.

### Software Requirements

Deep Learning Frameworks: Among the most well-known deep learning frameworks are:

TensorFlow v2.16.1: This is a resourceful general-purpose and widely used framework developed by Google.

Keras v3.5.0: High-level API for building and deploying deep learning models typically with TensorFlow.

PyTorch v2.4.0: Another highly versatile dynamic computation graph framework.

Python v3.12.5: Well, actually, Python is the programming core of deep learning. This version has been specified in regard to certain state-of-the-art deep learning frameworks and some libraries that concern them.

Image Processing Libraries: OpenCV v4.10.0, which is effectively the biggest and best-known library used to undertake almost all activities of image processing such as reading, writing, and manipulating images, which have also formed vital parts in deep learning applications.

## 3.7 Data Set

**Size:** This dataset, with its 5,249 images, makes a massive bedrock for building and testing learning models. For sample sizes in any field, that sizable an amount will generally amount short-of sufficient to build a very strong model, particularly when combined with data augmentation techniques.

**Image Format:** It is JPEG format an easy widely supported and efficient format to store and handle medical images. That really is a best compromise between image quality and file size.

**Resolution:** The 512 pixel x 512 pixel resolution provides a sensible level of detail pertinent to the brain tumour images. This resolution is used in many medical image analysis tasks and could always be modified as per specific model architectures.

**Labeling:** All images are annotated with bounding box. Bounding box annotation makes it very useful in training the object detection models as it defines a precise location and

dimension of the tumors in each image, making it learn to accurately localize tumors.

**Label Format:** The CSV file format is a simple, widely used format for storing tabular data. It provides very simple ways of linking image filenames with their corresponding bounding box coordinates (x,y,width,height).

#### **Skill Recommendation for Machine Learning:**

This suits a lot of characteristics of the dataset to performing different tasks in machine learning, which includes:

**Actual:** Tumor detection in the brain , Models are built to sort images as containing a tumor or not.

**Tumor localization:** Models of object detection intended to locate and demarcate the edges of tumors in images accurately.

**Tumor segmentation:** Image segmentation models have been trained to produce pixel-level masks of tumor areas in images.

### **3.8 Module Division**

#### **Ajith:**

Detecting Brain and Lung Cancer using ResNet-50 Data Acquisition

#### **Anna:**

Brain and Lung Cancer detection using Inception v3

#### **Arjun:**

Brain and Lung Cancer detection using VGG16

Pre-Processing of Data

#### **Athulya:**

Brain and Lung Cancer detection using YOLO v8.

### **3.9 Key Deliverables**

The pipeline has been established for machine learning by modern deep learning models: Inception-v3, ResNet-50, VGG-16, and YOLOv8 prototypes that have been built. This pipeline is focused on image analysis for medical practice, mainly MRI and CT images, to assist cancer diagnosis. The system offers a two-fold classification:

**Cancer Detection:** The primary task is to classify input images as 'cancerous' or 'non-cancerous' which implies the occurrence or absence of malignant tissue.

**Cancer Type Prediction:** In the presence of cancerous tissue, the system will classify the type of cancer and categorize it among either Glioma, Meningioma, or Pituitary, which is much informative for further decision-making during clinical intervention.

### 3.10 Gantt Chart

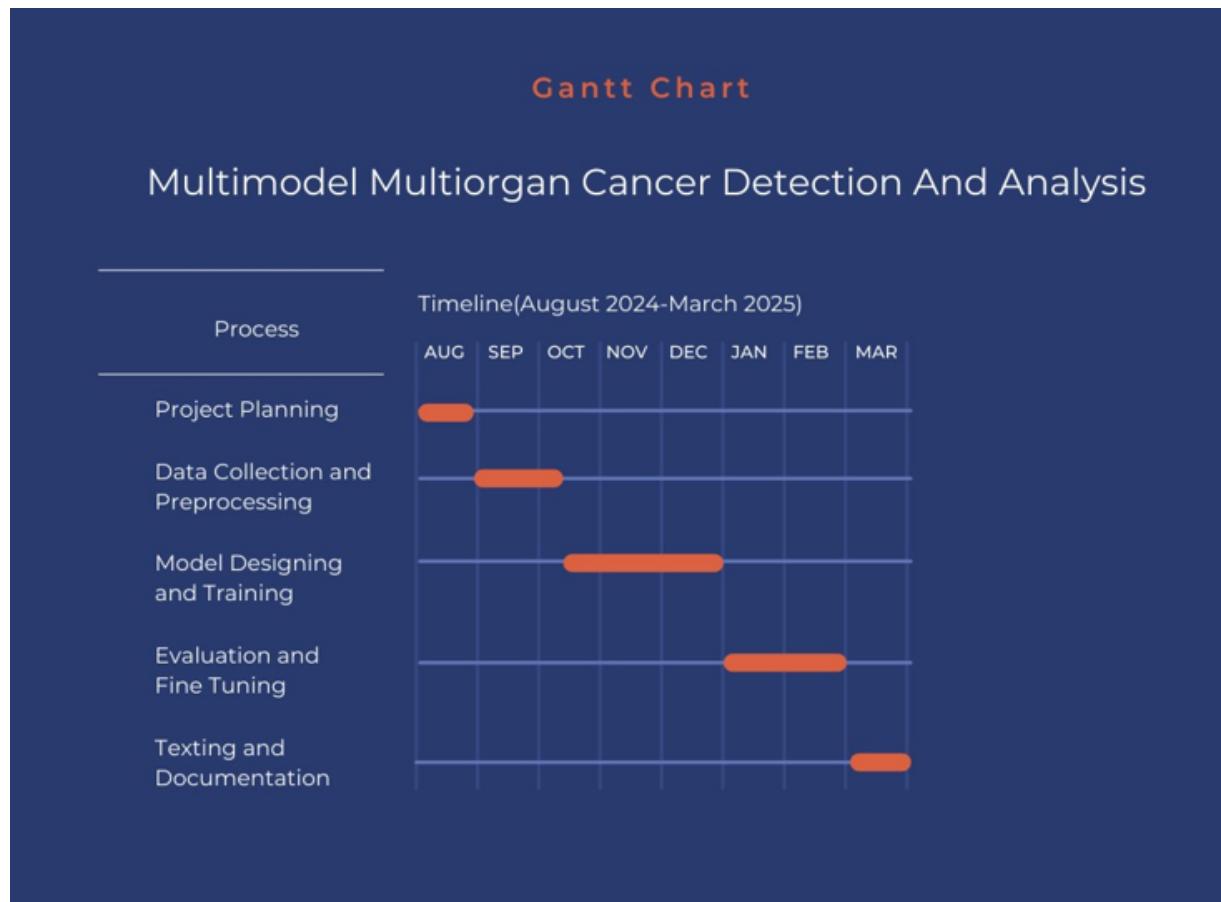


Figure 3.3: Gantt Chart

# Chapter 4

## Results and Discussions

This chapter presents the results obtained from our multi-model approach to brain and lung cancer detection using MRI and CT scans. The performance of YOLOv8, InceptionV3, VGG16, and ResNet50 is evaluated based on various metrics, including accuracy, precision, recall, and F1-score. A comparative analysis highlights the strengths and limitations of each model in detecting cancerous regions. Furthermore, the implications of these findings are discussed in relation to real-world medical applications, emphasizing the potential benefits and challenges of deep learning in cancer diagnosis.

### 4.1 Comparative Analysis

Model	Max Accuracy	Training Epochs	Convergence	Key Characteristics
ResNet-50	~98%	30	Moderate with volatility	High accuracy with extremely volatile validation, smooth training curve, significant validation loss spikes
VGG-16	~95%	30	Steady	Consistent performance between training and validation, good precision metrics, plateaus around epoch 30
InceptionV3	~98%	30	Fast with fluctuations	Highest accuracy, rapid initial increase, some validation drops around epoch 25, stabilizes after initial volatility
YOLO-V8	~95%	30	Fast	Comprehensive metrics tracking (mAP50, precision, recall), specialized for object detection, rapid loss decrease

Figure 4.1: Comparison between the 4 models for Brain Cancer Detection

Model	Max Accuracy	Training Epochs	Convergence	Key Characteristics
VGG-16	~85%	25	Moderate	Consistent improvement, validation stabilizes at ~85%
Inception-V3	~98%	22	Fast training	Large training-validation gap, step improvements
ResNet50	~99%	15	Very fast	Highest accuracy, validation fluctuations, near-zero loss
YOLOv8	~98%	50	Steady	Long training, consistent improvement, smooth loss curve

Figure 4.2: Comparison between the 4 models for Lung Cancer Detection

## 4.2 System Outcomes

- **Independent Model Performance:** Comparative evaluation of ResNet50, YOLOv8, VGG16, and InceptionV3 models in detecting brain and lung cancer. Performance measured in terms of accuracy, precision, recall, and F1-score.
- **Model-Specific Strengths:** Identification of which model performs best for MRI and CT scan analysis. YOLOv8 excels in object detection and localization. CNN-based architectures (ResNet50, VGG16, InceptionV3) provide strong classification performance.
- **Accuracy and Precision:** High performance across deep learning models for detecting tumors. Comparison of false positives and false negatives among different models.
- **Comparison Across Modalities:** Evaluation of how well each model detects cancer in different imaging modalities (MRI and CT scans). Insights into which model is more suited for brain vs. lung cancer detection.

- **Visualization and Interpretation:** Model-generated heatmaps or Grad-CAM visualizations for explainability. Bounding box predictions for YOLOv8, showcasing tumor localization.
- **Potential Deployment Readiness:** Assessment of computational efficiency and real-time applicability. Consideration of model optimization for clinical use.

In this chapter, we have explored the scope and discussions surrounding the multi-model cancer detection system. The comparative analysis of deep learning models has highlighted their effectiveness in detecting brain and lung cancer using MRI and CT scans. The discussions have also emphasized the need for continuous improvement in AI-driven medical diagnosis. By integrating advanced machine learning techniques with clinical workflows, this research paves the way for more accurate, interpretable, and deployable cancer detection solutions. The future scope outlined in this chapter underscores the potential for real-world applications and further optimizations that could enhance the model's performance and usability in healthcare environments.

# Chapter 5

## Conclusions & Future Scope

### 5.1 Conclusion

The multi-model approach using YOLOv8, InceptionV3, VGG16, and ResNet50 shows notably different performance patterns between brain and lung cancer detection tasks. For brain cancer detection, accuracy levels cluster around 95-98%, with ResNet-50 and InceptionV3 reaching ~98% accuracy. In contrast, lung cancer detection shows higher overall accuracy, with ResNet50 achieving an exceptional ~99% accuracy with only 15 training epochs - significantly fewer than required for other models.

Convergence patterns differ substantially between applications. Brain models all required 30 epochs, while lung models varied widely from 15 epochs (ResNet50) to 50 epochs (YOLOv8). This suggests tissue-specific characteristics may impact model learning efficiency. Notably, ResNet50 showed contradictory behavior - moderate convergence with volatility for brain imaging but very fast convergence for lung imaging.

InceptionV3 demonstrated high accuracy across both applications but with different characteristics - showing fluctuations in brain detection while exhibiting a training-validation gap in lung detection. YOLOv8 maintained consistent performance with fast convergence for brain detection and steady improvement over longer training for lung detection.

These comparative findings suggest that optimal model selection should be tissue-specific rather than universal. The dramatic improvement in ResNet50's performance on lung images indicates it may be particularly well-suited for lung cancer detection, while a more balanced approach might be needed for brain cancer detection where performance differences between models were less pronounced.

## 5.2 Future Scope

- **Integration with Clinical Workflows:** The model can be further optimized for real-world hospital settings by integrating it into radiology software for automated cancer detection assistance.
- **Larger and More Diverse Datasets:** Training on larger, more diverse datasets covering different demographics and cancer stages will improve model generalization and robustness.
- **Multi-Modal Data Fusion:** Combining MRI and CT scans with histopathological images or genomic data could enhance diagnostic accuracy.
- **Explainability and Interpretability:** Implementing techniques such as Grad-CAM to visualize model decisions will improve trust among medical professionals.
- **Lightweight and Edge Deployment:** Optimizing the models for mobile and edge devices can enable real-time cancer detection in resource-constrained environments.
- **Real-Time Tumor Tracking:** Extending the work to support real-time tracking of tumor progression across multiple scans over time.
- **Collaboration with Medical Experts:** Partnering with oncologists and radiologists to validate and fine-tune the model for clinical approval and FDA certification.

These future directions will help enhance the practical applicability of AI-driven cancer detection systems, making them more effective and widely accepted in medical practice.

# **Chapter 6**

## **Conclusion**

We conclude by including machine learning in applications of medical imaging, and it shows how dramatically it can change the methodologies of diagnostics. Preprocessing raw MRIs and CT scans, using complex neural networks for feature extraction and classification, has made the work easier in identifying such fatal conditions. Models such as ResNet50 and YOLOv8 give very good performance while precision and recall give a measure for the systems' competency in practical application.

However, much promise lies in the future by advancing stronger algorithms, a wider array of datasets, and complicated hardware development, as the field continues to mature. By connecting technology with medicine in such a way that these ML systems become one of the key components, medical practice will not only have better identifiers but may also revolutionize the conduct of medicine as personalized and pandemic preventive medicine in the future offers much promise.

## References

- [1] A. K. Swain, A. Swetapadma, J. K. Rout, and B. K. Balabantaray, “Classification of non-small cell lung cancer types using sparse deep neural network features,” *Biomedical Signal Processing and Control*, vol. 87, p. 105485, 2024. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1746809423009187>
- [2] M. S. Majib, M. M. Rahman, T. M. S. Sazzad, N. I. Khan, and S. K. Dey, “Vgg-scnet: A vgg net-based deep learning framework for brain tumor detection on mri images,” *IEEE Access*, vol. 9, pp. 116 942–116 952, 2021.
- [3] A. Younis, Q. Li, Z. Afzal, M. Jajere Adamu, H. Bello Kawuwa, F. Hussain, and H. Hussain, “Abnormal brain tumors classification using resnet50 and its comprehensive evaluation,” *IEEE Access*, vol. 12, pp. 78 843–78 853, 2024.
- [4] Q. Yao, D. Zhuang, Y. Feng, Y. Wang, and J. Liu, “Accurate detection of brain tumor lesions from medical images based on improved yolov8 algorithm,” *IEEE Access*, vol. 12, pp. 144 260–144 279, 2024.
- [5] F. J. Díaz-Pernas, M. Martínez-Zarzuela, M. Antón-Rodríguez, and D. González-Ortega, “A deep learning approach for brain tumor classification and segmentation using a multiscale convolutional neural network,” *Healthcare (Basel)*, vol. 9, no. 2, p. 153, 2021, published online: February 2, 2021. [Online]. Available: <https://doi.org/10.3390/healthcare9020153>

## **Appendix A: Presentation**

# MULTI-MODEL CANCER DETECTION

FINAL PRESENTATION

## CONTENTS

- 1) PROBLEM DEFINITION
- 2) PURPOSE AND NEED
- 3) OBJECTIVES
- 4) LITERATURE SURVEY
- 5) METHODOLOGY
- 6) ARCHITECTURE  
DIAGRAM
- 7) MODULES
- 8) MODULE EXPLANATION
- 9) ASSUMPTIONS
- 9) CHALLENGES
- 10) EXPECTED OUTPUT
- 11) SYSTEM  
REQUIREMENTS
- 12) WORK BREAKDOWN
- 13) GANTT CHART
- 14) RESULTS AND OUTPUT
- 15) CONCLUSION
- 16) REFERENCES

### Project Team

Ajith Varghese Abraham  
(U2103018)  
Anna Prince  
(U2103040)  
Arjun Martin  
(U2103047)  
Athulya S Pai  
(U2103055)

### Project Guide

Mr. Harikrishnan M  
Asst. Professor  
Dept. of Computer Science

## PROBLEM DEFINITION

Detecting cancer at an early stage is essential for improving treatment outcomes, particularly for conditions like lung cancer and brain tumors. Traditional diagnostic methods often suffer from inefficiencies, inaccuracies, and delays, making it necessary to explore advanced, automated solutions. This study focuses on comparing four machine learning models—YOLOv8, VGG16, Inception V3, and ResNet50—to assess their effectiveness in detecting lung cancer and brain tumors. By evaluating these models, we aim to identify the most accurate and reliable approach for automating the detection process, ultimately enhancing early diagnosis and patient care.

07-04-2025

MULTI-MODEL CANCER DETECTION

3

## PURPOSE AND NEED

- Detecting brain and lung cancer at an early stage is key to improving treatment success and increasing the chances of survival for patients.
- Traditional diagnostic methods are often time consuming and prone to error

07-04-2025

MULTI-MODEL CANCER DETECTION

4

## OBJECTIVES

- **Identify** and diagnose cancer of vital organs (Brain and Lungs) from diagnostic images (MRI and CT-Scan) using deep learning models **ResNet50, VGG16, Inception v3, and YOLO v8**.
- **Compare** the performance of various models to determine the most effective architecture for multi-organ cancer detection.

## LITERATURE SURVEY

PAPER	ADVANTAGES	DISADVANTAGES
<b>Brain tumor detection from images and comparison with transfer learning methods and 3-layer CNN. Mohammad Zafer Khaliki &amp; Muhammet Sinan Başarslan(2021)</b>	<ul style="list-style-type: none"><li>• Model Versatility</li><li>• Use of open source data</li></ul>	<ul style="list-style-type: none"><li>• Overfitting risk</li><li>• Not fully generalizable</li></ul>
<b>Vgg-senet: A VGG net-based deep learning framework for brain tumor detectionon MRI images.Mohammad shahjahan majib, md. MAHBUBUR RAHMAN , (member, IEEE),T. M. Shahriar sazzad, nafiz imtiaz khan , and samrat kumar dey(2022)</b>	<ul style="list-style-type: none"><li>• Hybrid Approach</li><li>• Comprehensive Comparison</li></ul>	<ul style="list-style-type: none"><li>• No tumor grading</li><li>• High computational requirements</li></ul>

# LITERATURE SURVEY

PAPER	ADVANTAGES	DISADVANTAGES
<b>Abnormal Brain Tumors Classification Using ResNet50 and Its Comprehensive Evaluation by Ayesha younis , Qiang li , Zargaam Afzal,Mohammed Jajere Adamu ,Halima Bello Kauwua , Fida Hussain and Hamid Hussain(2023)</b>	<ul style="list-style-type: none"> <li>Transfer Learning</li> <li>Improved Generalization</li> </ul>	<ul style="list-style-type: none"> <li>Model interpretability</li> <li>Limited dataset</li> </ul>
<b>Object Detection for Brain Cancer Detection and Localization by Francesco Mercaldo ,Luca Brunese ,Fabio Martinelli ,Antonella Santone and Mario Cesarelli(2023)</b>	<ul style="list-style-type: none"> <li>High precision and recall</li> <li>Effective Localization</li> </ul>	<ul style="list-style-type: none"> <li>Smaller dataset</li> <li>Complex setup</li> </ul>

07-04-2025

MULTI-MODEL CANCER DETECTION

7

# LITERATURE SURVEY

PAPER	ADVANTAGES	DISADVANTAGES
<b>Classifying Brain Tumors on Magnetic Resonance Imaging by Using Convolutional Neural Networks by Marco Antonio Gómez-Guzmán , Laura Jiménez-Beristáin , Enrique Efren García-Guerrero,Oscar Roberto López-Bonilla, Ulises Jesús Tamayo-Perez , José Jaime Esqueda-Elizondo, Kenia Palomino-Vizcaino and Everardo Inzunza-González(2020)</b>	<ul style="list-style-type: none"> <li>Use of multiple CNN models</li> <li>Preprocessing techniques</li> </ul>	<ul style="list-style-type: none"> <li>Overfitting risk</li> <li>Single modularity focus</li> </ul>

07-04-2025

MULTI-MODEL CANCER DETECTION

8

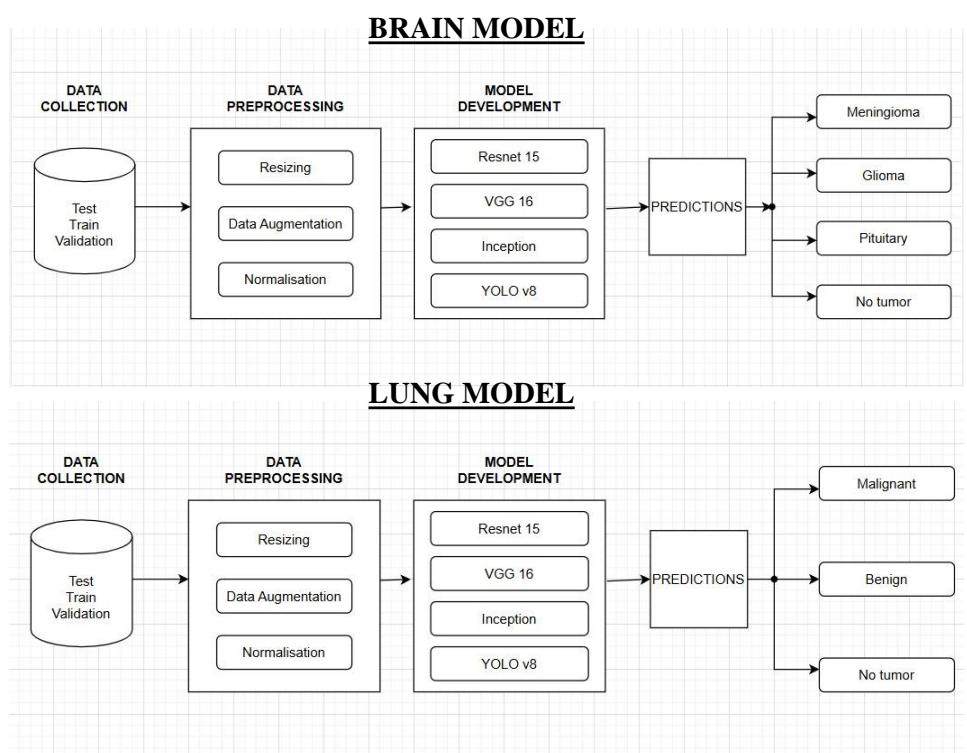
- ## METHODOLOGY
- Uses types of Convolutional Neural Networks.
  - Key architectures :
    - ❖ ResNet50
    - ❖ Inception v3
    - ❖ VGG16
    - ❖ YOLO v8

07-04-2025

MULTI-MODEL CANCER DETECTION

9

## ARCHITECTURE DIAGRAM



07-04-2025

MULTI-MODEL CANCER DETECTION

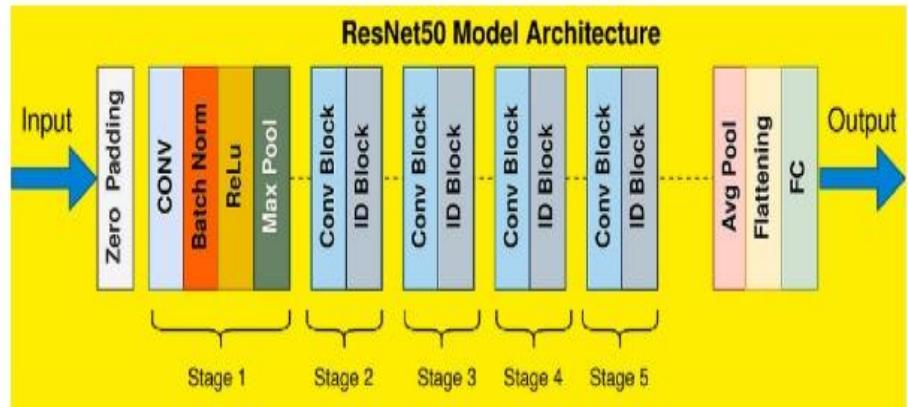
10

# MODULES

- ResNet50
- Inception v3
- VGG16
- YOLO v8

07-04-2025 MULTI-MODEL CANCER DETECTION 12

# RESNET-50



07-04-2025 MULTI-MODEL CANCER DETECTION 13

# RESNET-50

**Title:** Modified ResNet50 for Brain Tumor Classification

## Key Features:

- Trained without pre-trained weights – Model learns entirely from medical data
- Custom top layers – Includes GlobalAveragePooling, Dense(256, ReLU), and Dropout layers.
- Dual Dropout Regularization – Helps reduce overfitting on a relatively small dataset.
- Aggressive Data Augmentation – Random rotation, flip, zoom, and shifts for better generalization.
- Adaptive Learning Rate Scheduling – Uses ReduceLROnPlateau to fine-tune training dynamics.
- Early Stopping – Monitors validation loss to stop training at the right time and restore best weights.
- 4-Class Output – Tailored specifically to brain tumor subtypes.

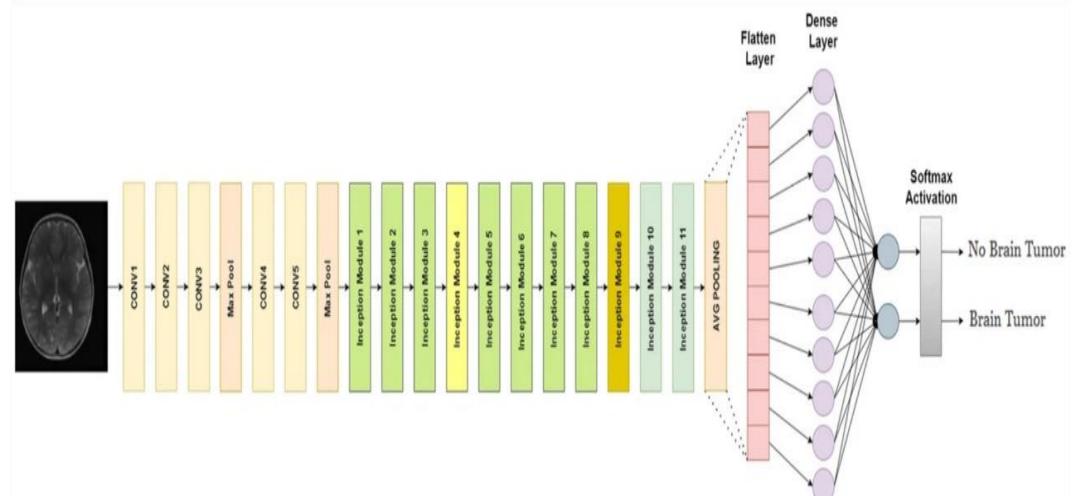
# RESNET-50

**Title:** Modified ResNet50 for Lung Cancer Classification

## Key Features:

- Uses pre-trained ImageNet weights – Benefits from learned features like edges and textures.
- Modified final FC layer – Replaced with new layer matching the number of lung cancer classes.
- Standard Data Normalization – Matches pre-trained model expectations with mean/std normalization.
- Lightweight Data Augmentation – Includes horizontal flips and small rotations.
- Fine-tuning entire model – Not just head; even pretrained layers get updated.
- Macro Precision & Recall Tracking – Ensures fair performance across all classes.
- Model Checkpointing – Best-performing model saved automatically based on validation accuracy.

# INCEPTION-v3



07-04-2025

MULTI-MODEL CANCER DETECTION

16

# INCEPTION-v3

**Title:** Modified InceptionV3 for Brain Tumor Classification

## Key Features:

- Custom InceptionV3 architecture without pre-trained weights, allowing full learning from brain tumor data.
- Extensive data augmentation including rotation, shifting, shearing, zooming, and horizontal flipping to improve generalization.
- A simplified classification head with GlobalAveragePooling, a dense layer of 1024 units (ReLU), and a softmax output layer for 4 tumor classes.
- Uses categorical crossentropy loss and tracks accuracy, precision, and recall for detailed performance evaluation.
- Designed to classify four categories: Glioma, Meningioma, Pituitary, and No Tumor.

07-04-2025

MULTI-MODEL CANCER DETECTION

17

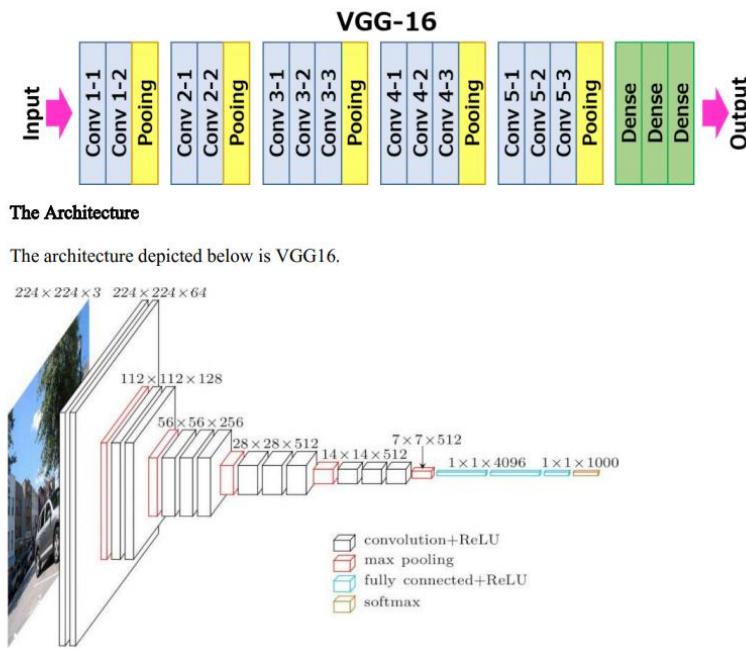
# INCEPTION-v3

**Title:** InceptionV3 for Lung Cancer Detection

## Key Features:

- Utilizes a pre-trained InceptionV3 base model (ImageNet weights) to transfer learned features to the lung cancer dataset.
- Fine-tunes only the top layers (after layer 249), while keeping the lower layers frozen for stable feature extraction.
- Data augmentation techniques like shifting, rotating, shearing, zooming, and flipping are used to enhance dataset diversity.
- A custom classifier head includes GlobalAveragePooling, a 512-unit dense layer with ReLU activation, followed by dropout (0.5), and a softmax layer.
- Employs EarlyStopping and ReduceLROnPlateau callbacks to prevent overfitting and adapt the learning rate during training.

# VGG-16



# VGG-16

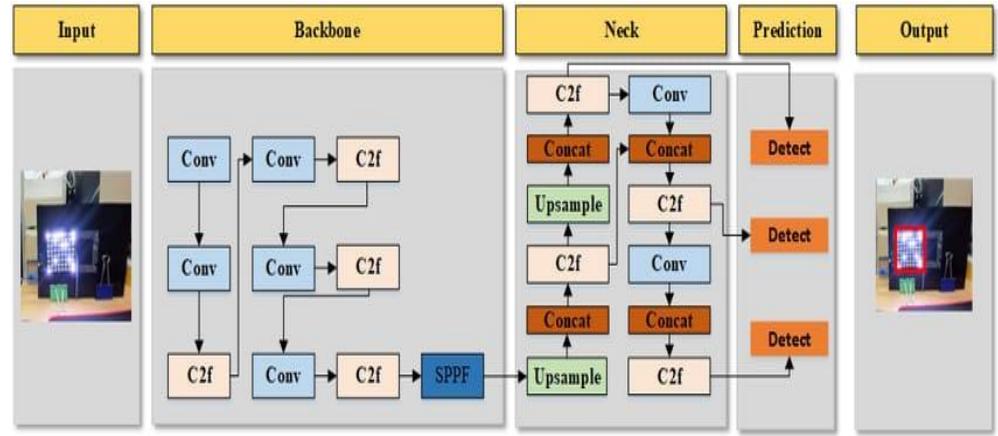
- **Title:** VGG16 for Brain Tumor Classification
- **Key Features:**
  - Utilizes pre-trained VGG16 weights (ImageNet) — Leverages transfer learning to speed up convergence.
  - Custom Top Layers – Includes Flatten, Dense(256, ReLU), Dropout(0.5), and final Dense(4, Softmax).
  - Freeze Base Layers – Prevents training of VGG16 convolutional base to retain learned features.
  - Data Augmentation – Random rotation, flips, zoom, shear, and shifts to improve generalization.
  - Adaptive Learning Rate – Uses ReduceLROnPlateau to adjust learning rate based on validation loss.
  - Early Stopping – Stops training early and restores best weights to prevent overfitting.
  - 4-Class Output – Tailored to classify Glioma, Meningioma, No Tumor, and Pituitary tumors.

# VGG-16

- Title :**VGG16-Based Transfer Learning for Lung Cancer Classification
- **Key Features:**

- Pretrained Base Model – Utilizes VGG16 with ImageNet weights; all convolutional layers are frozen for feature extraction.
- Custom Classification Head – Includes Flatten, Dense(512, ReLU), Dropout(0.5), and a final Dense layer with SoftMax activation for 3-class output.
- Data Preprocessing – Images are rescaled to [0,1] using ImageDataGenerator; no complex augmentations used.
- Image Size – All images resized to 224×224 to match VGG16 input requirements.
- Training Configuration –
  - Optimizer: Adam (lr=0.0001)
  - Loss: Categorical Crossentropy
  - Metrics: Accuracy
  - Epochs: 10
  - Batch Size: 32

# YOLO-v8



# YOLO-v8

**Title:** YOLOv8 for Brain Tumor Detection using Object Localization

## Key Features:

- Pretrained YOLOv8 Model – Leverages Ultralytics' YOLOv8 with pretrained weights for object detection tasks.
- Bounding Box-Based Detection – Detects and localizes tumor regions within MRI/CT scans.
- Custom Annotations – Trained on manually labeled data (bounding boxes) tailored for tumor types.
- High-Speed Inference – Real-time detection capability suitable for clinical screening pipelines.
- Flexible Architecture – Scalable from small to large models (YOLOv8n to YOLOv8x) based on resource constraints.
- Robust Data Augmentation – Includes mosaic, HSV adjustments, flipping, and scaling to improve generalization.
- Multi-Class Detection – Capable of identifying and distinguishing between multiple tumor types.

## YOLO-v8

**Title:** YOLOv8 for Lung Cancer Detection using CT Scans

### Key Features:

- YOLOv8 Object Detection Framework – Utilizes Ultralytics' state-of-the-art real-time detector for medical imaging.
- Lung Nodule Localization – Detects and outlines cancerous nodules using bounding boxes on CT scan slices.
- Custom Labeled Dataset – Trained on annotated CT images with class-specific tumor regions (e.g., benign, malignant).
- Augmentation Pipeline – Incorporates brightness shifts, rotation, flipping, and scale jittering for robust learning.
- Transfer Learning – Fine-tuned from pretrained COCO weights to adapt to lung cancer domain.
- Compact & Efficient – Supports lightweight deployment for edge inference or hospital systems.
- Multi-Class Detection – Supports classification of multiple lung cancer types from localized regions.

## ASSUMPTIONS

- ❖ **Data Availability:** Sufficient and diverse MRI/CT scans for training
- ❖ **Standardized Image Quality:** Consistency across different medical institutions
- ❖ Fine-tuning of pretrained models will be efficient

## CHALLENGES

- ❖ **Data Collection:** Difficulty in acquiring high-quality, annotated MRI/CT scans.
- ❖ **Data Preprocessing:** Variations in image resolution and quality may introduce noise.
- ❖ **Computational Complexity:** High resource demand for training large models
- ❖ **Regulatory and Ethical Issues:** Compliance with medical standards and privacy laws

## EXPECTED OUTPUT

- ❖ **Independent Model Performance:** Comparative evaluation of ResNet50, YOLOv8, VGG16, and Inception V3 models in detecting brain and lung cancer
- ❖ **Model-Specific Strengths:** Insights into which model performs best for MRI and CT scan analysis for each type of cancer (brain or lung)
- ❖ **Accuracy and Precision:** Expected high performance in terms of accuracy, precision, recall, and other metrics for cancer detection in each model

# SYSTEM REQUIREMENTS

## HARDWARE

- CPU - Intel Core i5, AMD Ryzen 5 or higher
- GPU - NVIDIA GTX 1060, AMD Radeon RX 6800 XT or higher
- RAM - 16gb or higher
- SSD - Depending on size of dataset

## SOFTWARE

- Deep Learning frameworks
  - Tensorflow v2.16.1
  - Keras v3.5.0
  - PyTorch v2.4.0
- Python v3.12.5
- Image Processing Libraries
  - OpenCV v4.10.0

07-04-2025

MULTI-MODEL CANCER DETECTION

28

# PROJECT WORK BREAKDOWN

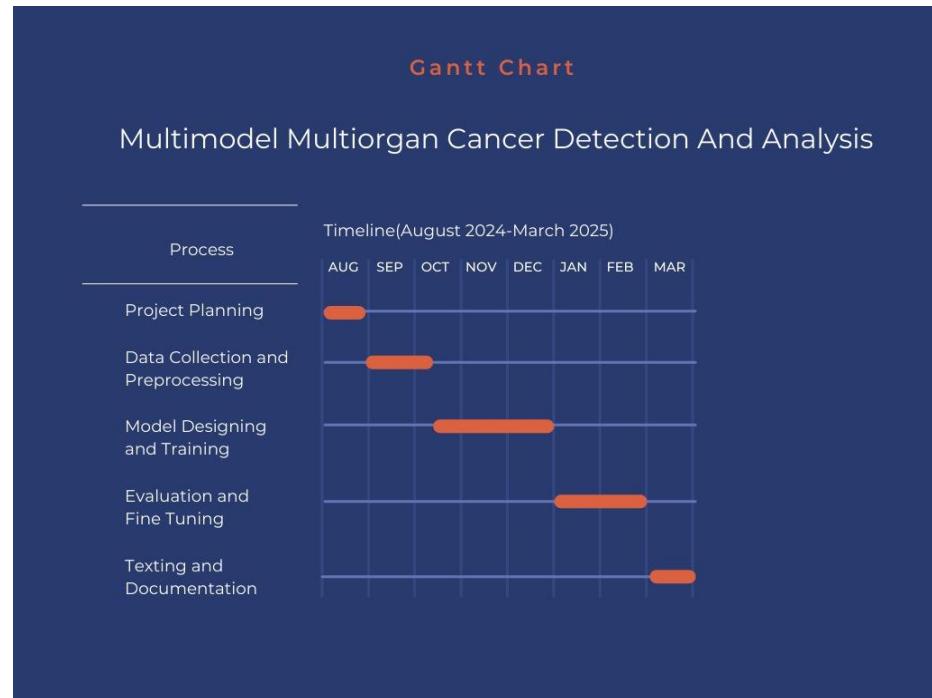


07-04-2025

MULTI-MODEL CANCER DETECTION

29

# GANTT CHART

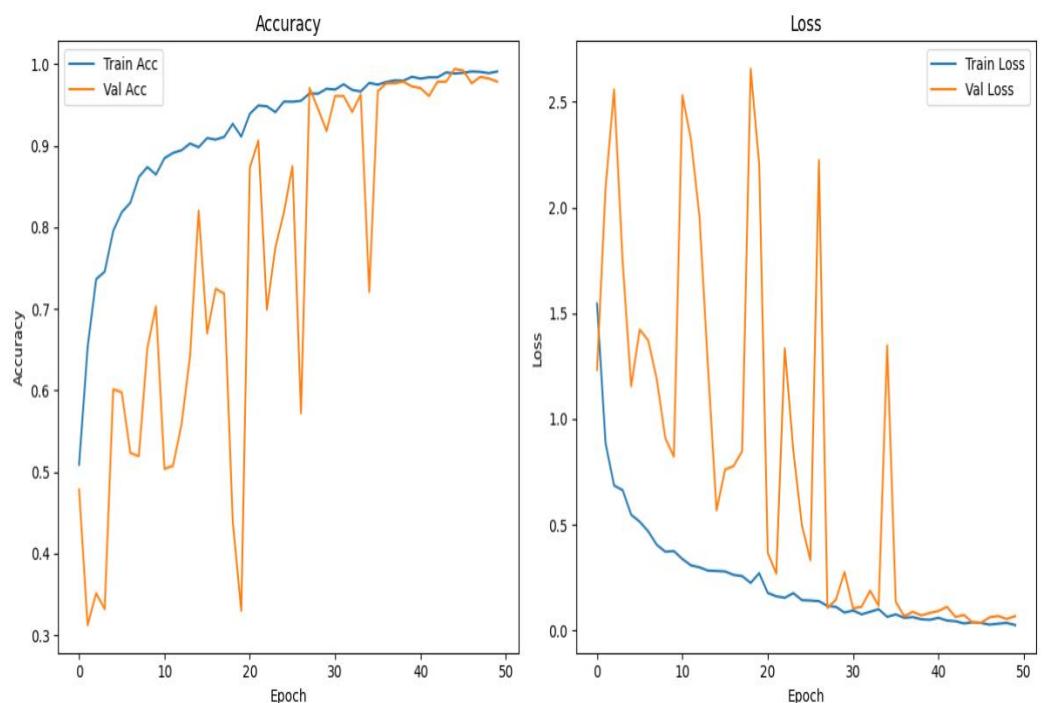


07-04-2025

30

# RESULTS

## RESNET-50 (BRAIN)



07-04-2025

31

MULTI-MODEL CANCER DETECTION

## RESULTS

### RESNET-50 (BRAIN)



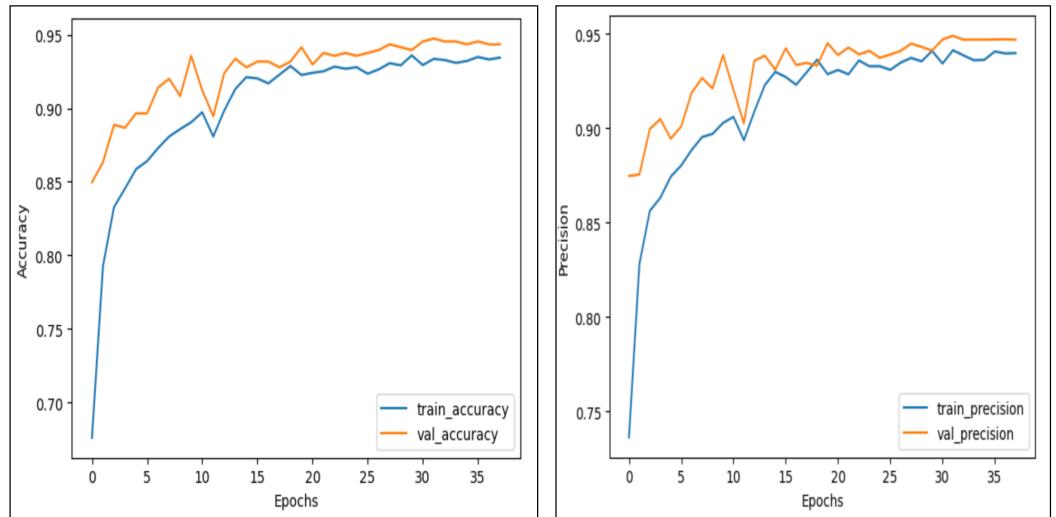
07-04-2025

MULTI-MODEL CANCER DETECTION

32

## RESULTS

### VGG-16 (BRAIN)



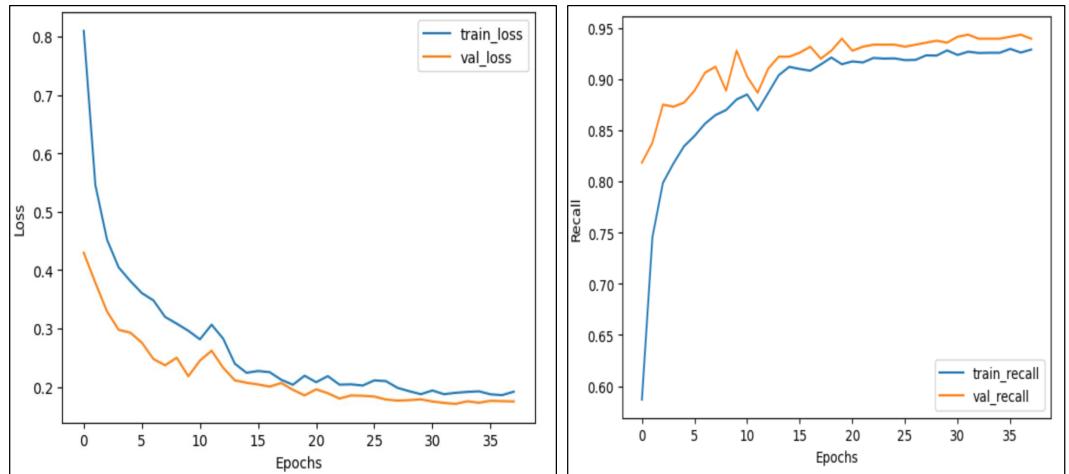
07-04-2025

MULTI-MODEL CANCER DETECTION

33

## RESULTS

### VGG-16 (BRAIN)



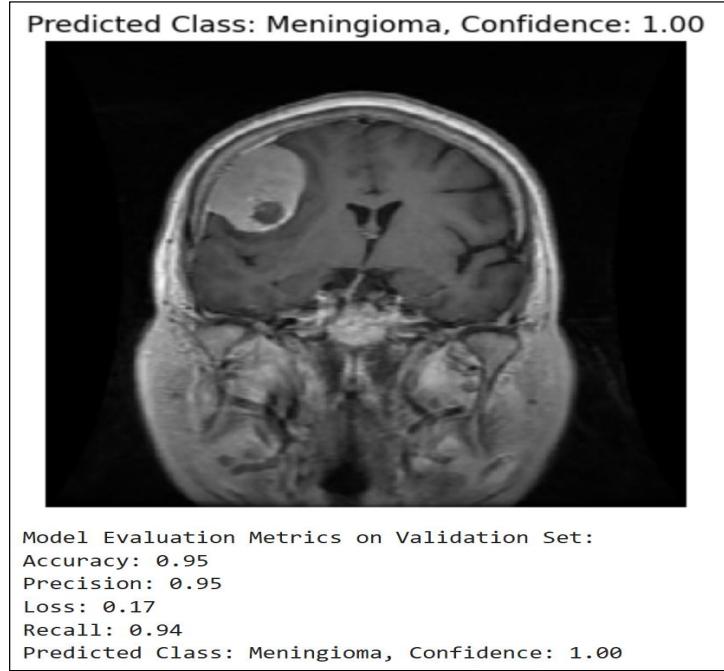
07-04-2025

MULTI-MODEL CANCER DETECTION

34

## RESULTS

### VGG-16 (BRAIN)



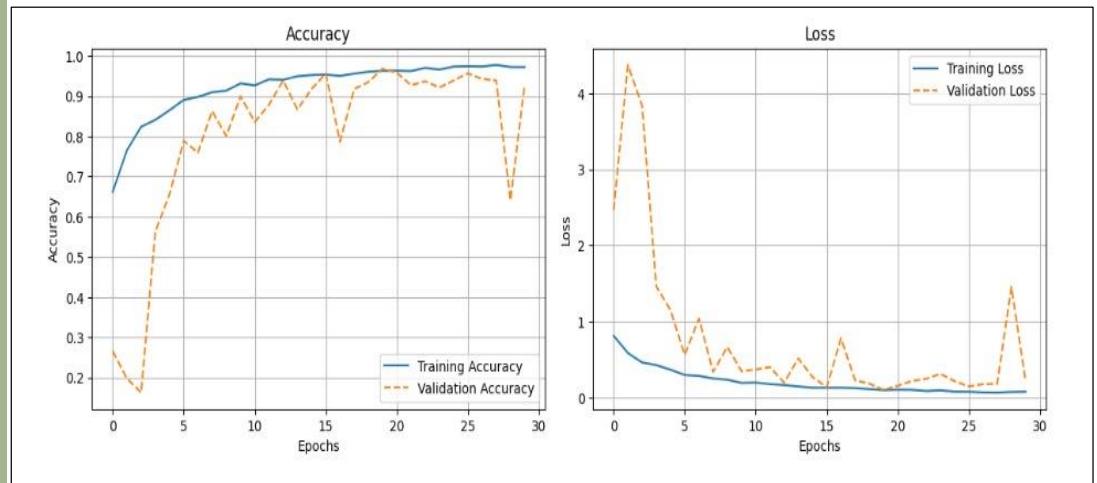
07-04-2025

MULTI-MODEL CANCER DETECTION

35

## RESULTS

### INCEPTION V3 (BRAIN)



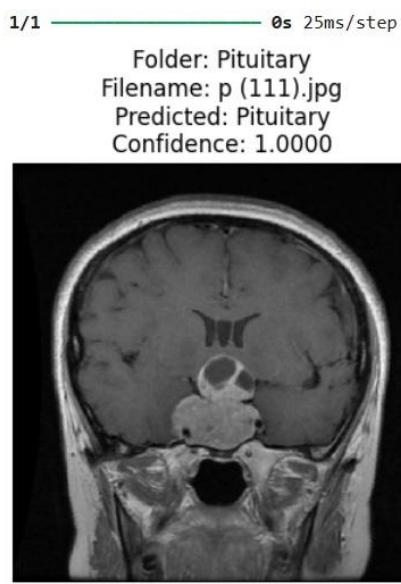
07-04-2025

MULTI-MODEL CANCER DETECTION

36

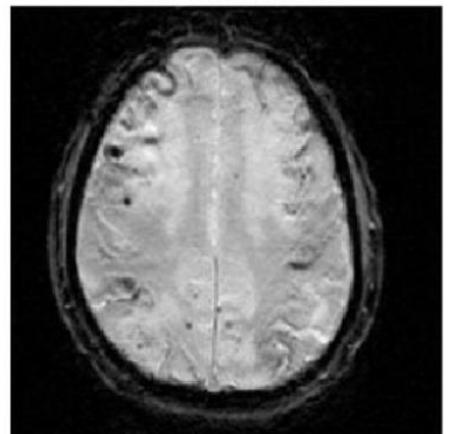
## RESULTS

### INCEPTION V3 (BRAIN)



1/1 ————— 0s 24ms/step

Folder: No Tumor  
Filename: image (53).jpg  
Predicted: No Tumor  
Confidence: 0.9999



07-04-2025

MULTI-MODEL CANCER DETECTION

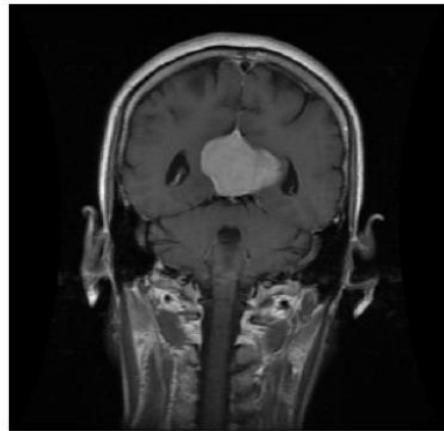
37

# RESULTS

## INCEPTION V3 (BRAIN)

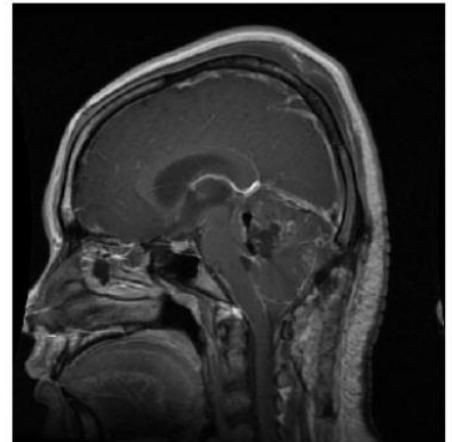
1/1 ————— 0s 24ms/step

Folder: Meningioma  
Filename: m (30).jpg  
Predicted: Meningioma  
Confidence: 0.9995



1/1 ————— 5s 5s/step

Folder: Glioma  
Filename: gg (167).jpg  
Predicted: Glioma  
Confidence: 0.9998



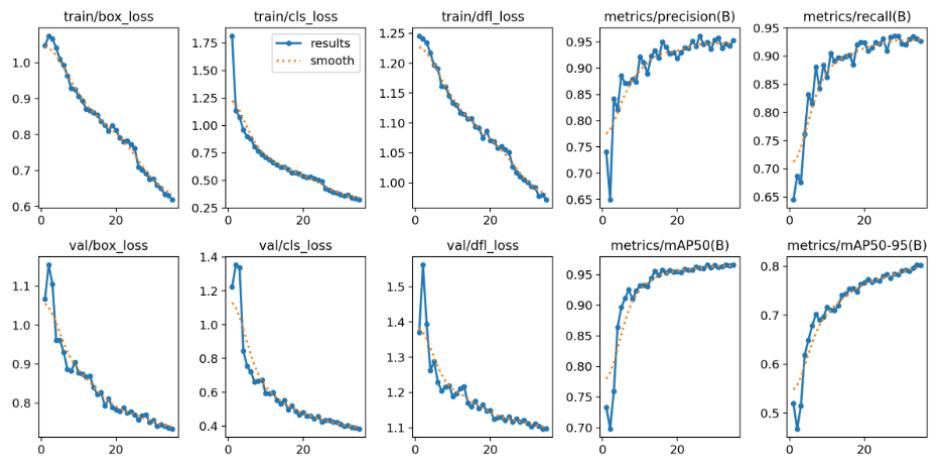
07-04-2025

MULTI-MODEL CANCER DETECTION

38

# RESULTS

## YOLO-V8 (BRAIN)



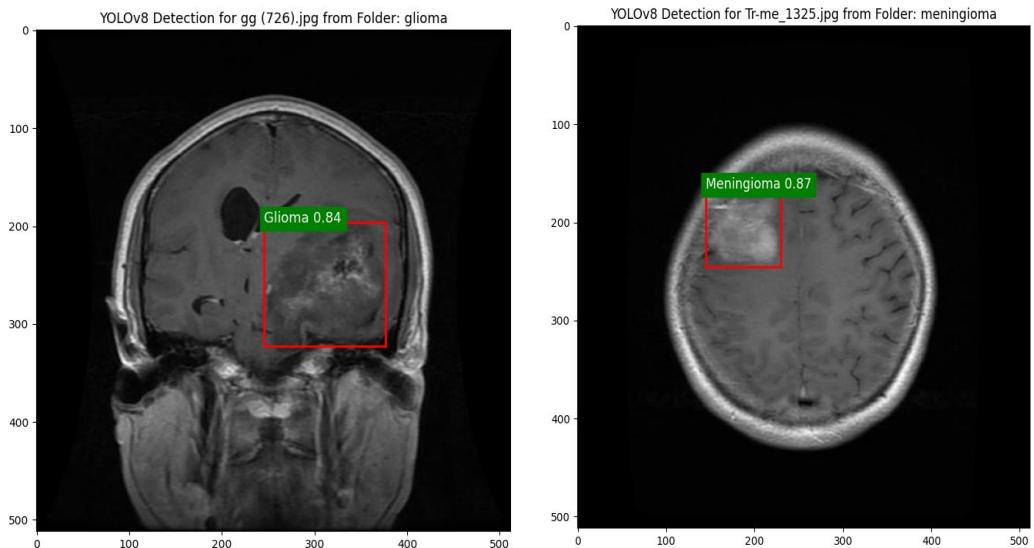
07-04-2025

MULTI-MODEL CANCER DETECTION

39

## RESULTS

### YOLO-V8 (BRAIN)



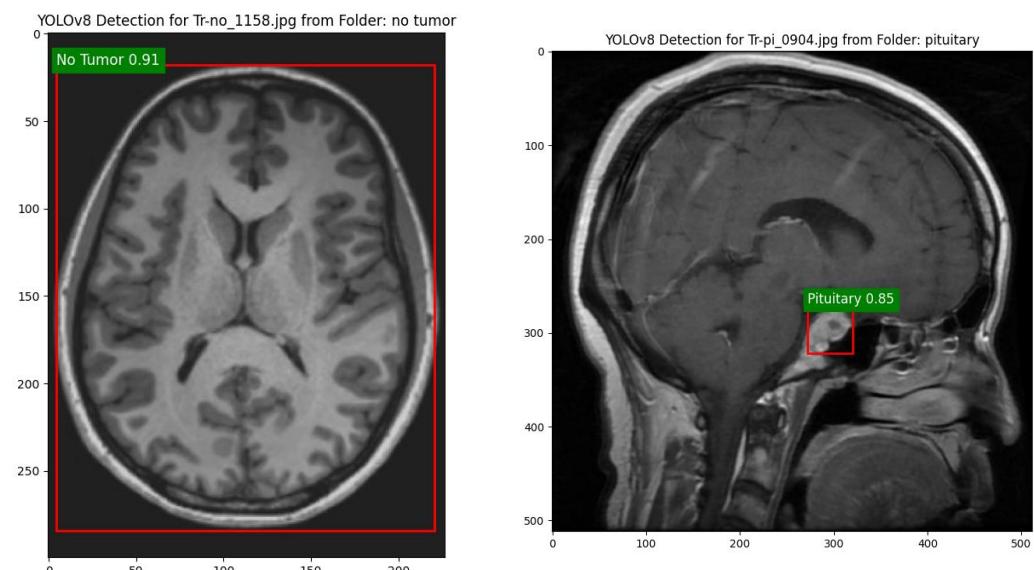
07-04-2025

MULTI-MODEL CANCER DETECTION

40

## RESULTS

### YOLO-V8 (BRAIN)



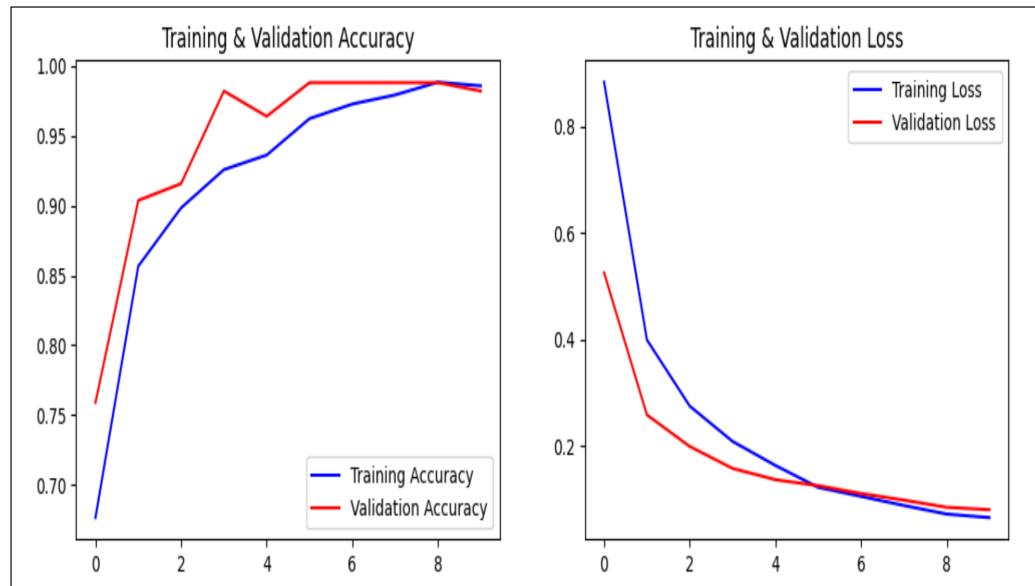
07-04-2025

MULTI-MODEL CANCER DETECTION

41

## RESULTS

### VGG-16 (LUNGS)



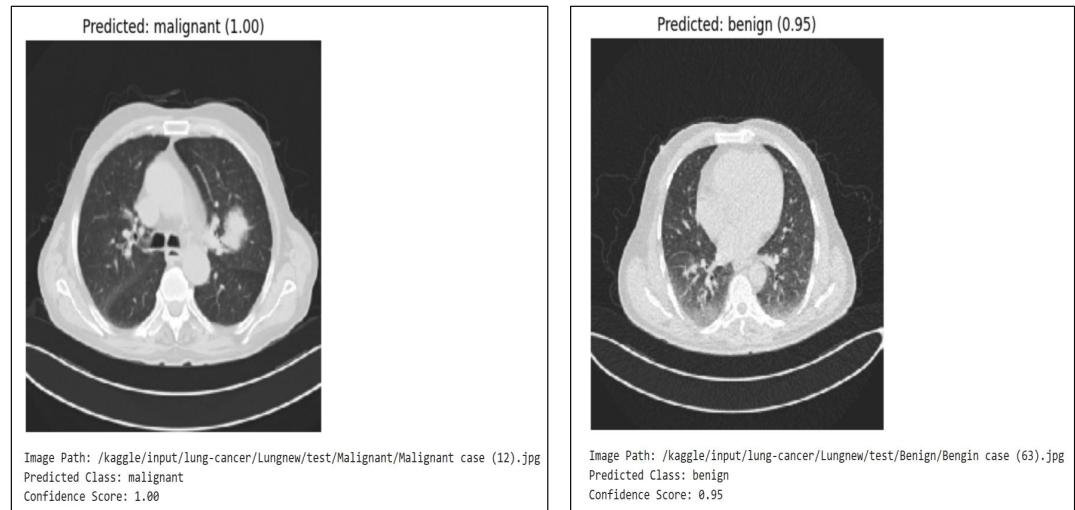
07-04-2025

MULTI-MODEL CANCER DETECTION

42

## RESULTS

### VGG-16 (LUNGS)



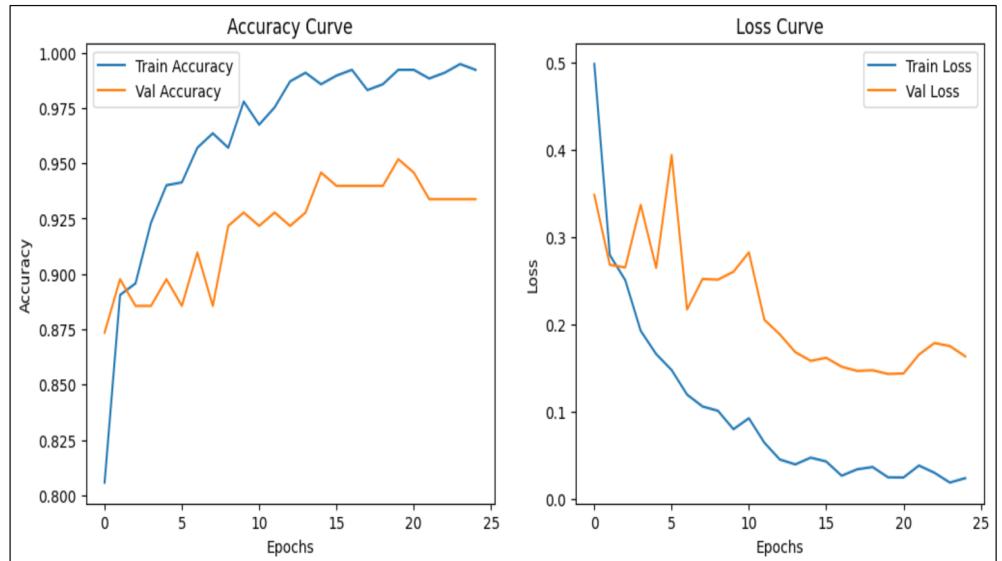
07-04-2025

MULTI-MODEL CANCER DETECTION

43

## RESULTS

### INCEPTIONV3 (LUNGS)



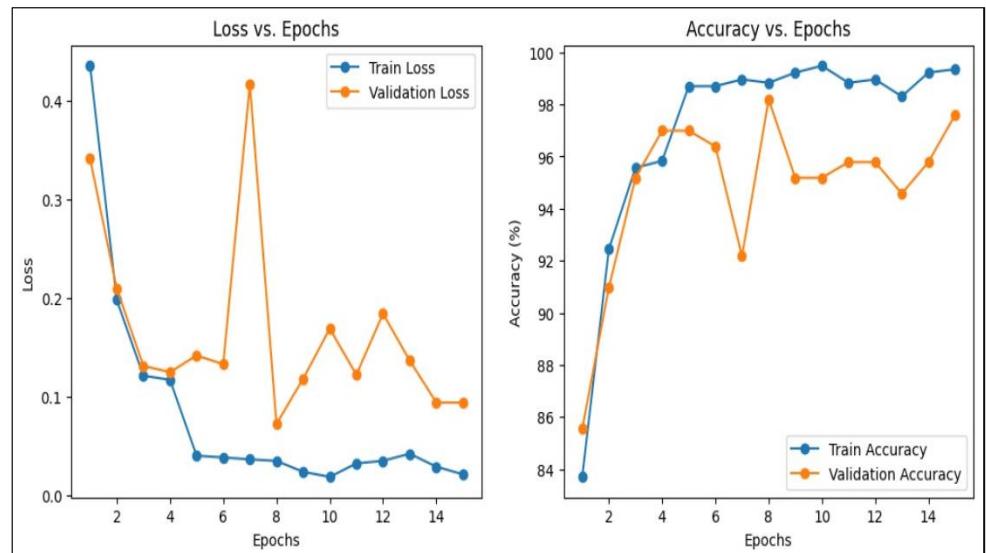
07-04-2025

MULTI-MODEL CANCER DETECTION

44

## RESULT

### RESNET 50 (LUNGS)



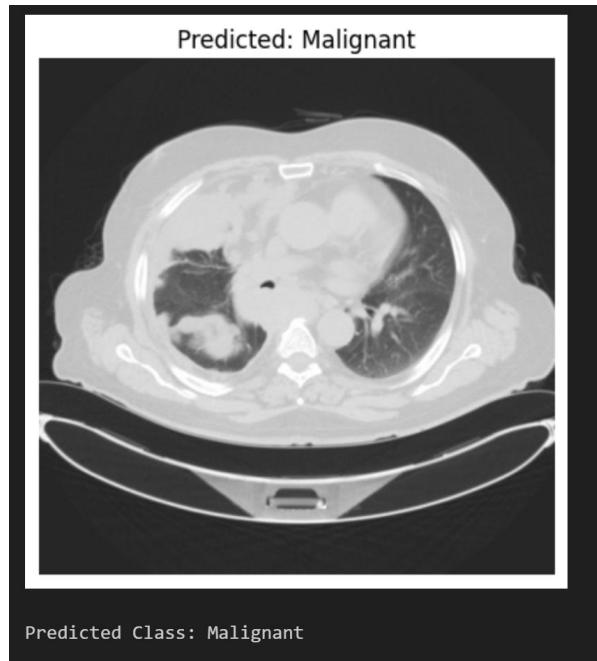
07-04-2025

MULTI-MODEL CANCER DETECTION

45

## RESULT

### RESNET 50 (LUNGS)



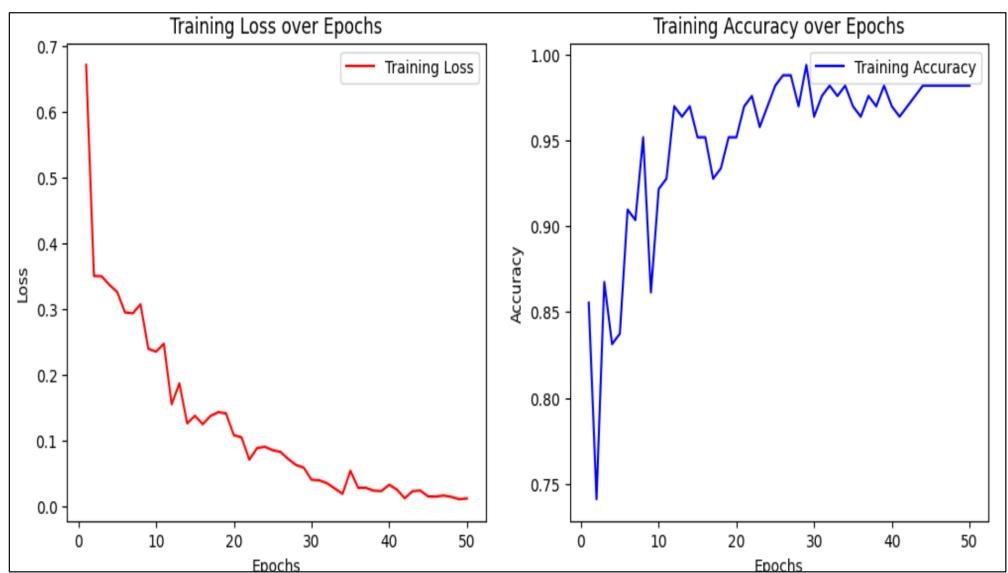
07-04-2025

MULTI-MODEL CANCER DETECTION

46

## RESULT

### YOLO V8 (LUNGS)



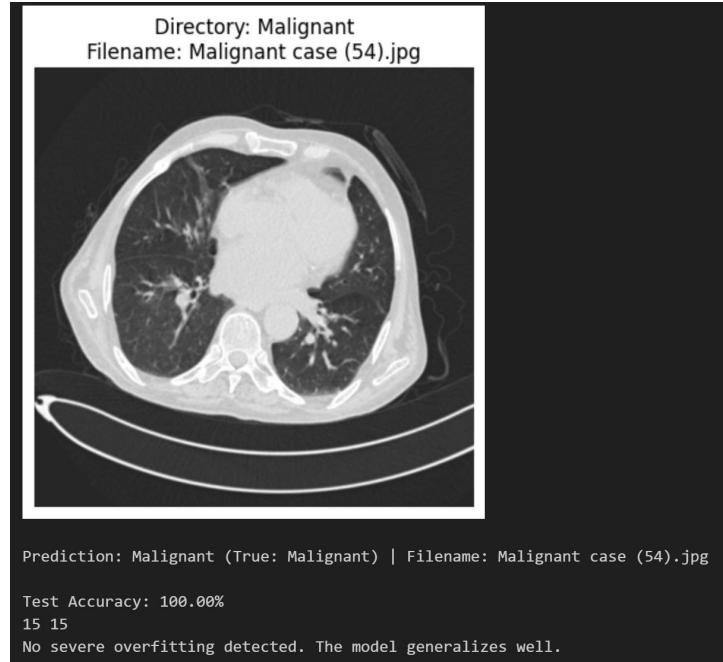
07-04-2025

MULTI-MODEL CANCER DETECTION

47

# RESULT

## YOLO V8 (LUNGS)



07-04-2025

MULTI-MODEL CANCER DETECTION

48

# COMPARISON

## BRAIN

Model	Max Accuracy	Training Epochs	Convergence	Key Characteristics
ResNet-50	~98%	30	Moderate with volatility	High accuracy with extremely volatile validation, smooth training curve, significant validation loss spikes
VGG-16	~95%	30	Steady	Consistent performance between training and validation, good precision metrics, plateaus around epoch 30
Inception V3	~98%	30	Fast with fluctuations	Highest accuracy, rapid initial increase, some validation drops around epoch 25, stabilizes after initial volatility
YOLO-V8	~95%	30	Fast	Comprehensive metrics tracking (mAP50, precision, recall), specialized for object detection, rapid loss decrease

07-04-2025

MULTI-MODEL CANCER DETECTION

49

# COMPARISON

## LUNGS

Model	Max Accuracy	Training Epochs	Convergence	Key Characteristics
VGG-16	~85%	25	Moderate	Consistent improvement, validation stabilizes at ~85%
Inception-V3	~98%	22	Fast training	Large training-validation gap, step improvements
ResNet50	~99%	15	Very fast	Highest accuracy, validation fluctuations, near-zero loss
YOLOv8	~98%	50	Steady	Long training, consistent improvement, smooth loss curve

07-04-2025

MULTI-MODEL CANCER DETECTION

50

# INFERENCE

**For Brain Cancer Detection:** Inception V3 is the best performing model with a maximum accuracy of approximately 98%. It achieved this in just 30 epochs, showing fast convergence despite some instability in validation metrics. The model demonstrates excellent learning capability for brain imaging features.

**For Lung Cancer Detection:** ResNet50 is clearly the best performer with a remarkable ~99% training accuracy and ~97% validation accuracy. It achieved these impressive results in only 15 epochs, making it both highly accurate and computationally efficient for lung cancer detection. The model showed very low final training loss (near zero) and despite some validation fluctuations, it maintained superior performance overall.

07-04-2025

MULTI-MODEL CANCER DETECTION

51

## CONCLUSION

This project **identifies and diagnoses** cancer in vital organs (Brain and Lungs) using medical diagnostic images (MRI and CT-Scans) and also **compares the performance** of various models to determine the most effective architecture for multi-organ cancer detection.

## REFERENCES

1. *"Classification of non-small cell lung cancer types using sparse deep neural network features"*, Biomedical Signal Processing and Control by Anil Kumar Swain, Aleena Swetapadma, Jitendra Kumar Rout, Bunil Kumar Balabantary, Volume 87, Part A,2024
2. Hollon T.C., Pandian B., Adapa A.R., Urias E., Save A.V., Khalsa S.S.S., Eichberg D.G., D'Amico R.S., Farooq Z.U., Lewis S., etal. *"Near real-time intraoperative brain tumor diagnosis using stimulated raman histology and deep neural networks"*. NatMed. 2020;26(1):52–58.
3. Rea G, et al. *"Beyond visual interpretation: quantitative analysis and artificial intelligence in interstitial lung disease diagnosis expanding horizons in radiology"*. Diagnostics. 2023;13:2333
4. Díaz-Pernas FJ, Martínez-Zarzuela M, Antón-Rodríguez M, González-Ortega D. *"A deep learning approach for brain tumor classification and segmentation using a multiscale convolutional neural network"*. InHealthcare 2021 Feb 2(Vol. 9, No. 2, p. 153). MDPI
5. R. P.R., R. A. S. Nair and V. G., *"A Comparative Study of Brain Cancer Detection using Machine Learning Algorithms,"* 2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT), Coimbatore, India, 2019, pp. 1-4, doi: 10.1109/ICECCT.2019.8869001.

## **Appendix B: Vision, Mission, Programme Outcomes and Course Outcomes**

# **Vision, Mission, Programme Outcomes and Course Outcomes**

## **Institute Vision**

To evolve into a premier technological institution, moulding eminent professionals with creative minds, innovative ideas and sound practical skill, and to shape a future where technology works for the enrichment of mankind.

## **Institute Mission**

To impart state-of-the-art knowledge to individuals in various technological disciplines and to inculcate in them a high degree of social consciousness and human values, thereby enabling them to face the challenges of life with courage and conviction.

## **Department Vision**

To become a centre of excellence in Computer Science and Engineering, moulding professionals catering to the research and professional needs of national and international organizations.

## **Department Mission**

To inspire and nurture students, with up-to-date knowledge in Computer Science and Engineering, ethics, team spirit, leadership abilities, innovation and creativity to come out with solutions meeting societal needs.

## **Programme Outcomes (PO)**

Engineering Graduates will be able to:

**1. Engineering Knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

**2. Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

**3. Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

- 4. Conduct investigations of complex problems:** Use research-based knowledge including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
- 5. Modern Tool Usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
- 6. The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal, and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
- 7. Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
- 8. Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
- 9. Individual and Team work:** Function effectively as an individual, and as a member or leader in teams, and in multidisciplinary settings.
- 10. Communication:** Communicate effectively with the engineering community and with society at large. Be able to comprehend and write effective reports documentation. Make effective presentations, and give and receive clear instructions.
- 11. Project management and finance:** Demonstrate knowledge and understanding of engineering and management principles and apply these to one's own work, as a member and leader in a team. Manage projects in multidisciplinary environments.
- 12. Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change.

## **Programme Specific Outcomes (PSO)**

A graduate of the Computer Science and Engineering Program will demonstrate:

### **PSO1: Computer Science Specific Skills**

The ability to identify, analyze and design solutions for complex engineering problems in multidisciplinary areas by understanding the core principles and concepts of computer science and thereby engage in national grand challenges.

### **PSO2: Programming and Software Development Skills**

The ability to acquire programming efficiency by designing algorithms and applying standard practices in software project development to deliver quality software products meeting the demands of the industry.

### **PSO3: Professional Skills**

The ability to apply the fundamentals of computer science in competitive research and to develop innovative products to meet the societal needs thereby evolving as an eminent researcher and entrepreneur.

## **Course Outcomes (CO)**

**Course Outcome 1:** Model and solve real world problems by applying knowledge across domains (Cognitive knowledge level: Apply).

**Course Outcome 2:** Develop products, processes or technologies for sustainable and socially relevant applications (Cognitive knowledge level: Apply).

**Course Outcome 3:** Function effectively as an individual and as a leader in diverse teams and to comprehend and execute designated tasks (Cognitive knowledge level: Apply).

**Course Outcome 4:** Plan and execute tasks utilizing available resources within timelines, following ethical and professional norms (Cognitive knowledge level: Apply).

**Course Outcome 5:** Identify technology/research gaps and propose innovative/creative solutions (Cognitive knowledge level: Analyze).

**Course Outcome 6:** Organize and communicate technical and scientific findings effectively in written and oral forms (Cognitive knowledge level: Apply).

## **Appendix C: CO-PO-PSO Mapping**

## Course Outcomes

After completion of the course the student will be able to:

SL.NO	Description	Bloom's Taxonomy Level
CO1	Model and solve real-world problems by applying knowledge across domains.	Level 3: Apply
CO2	Develop products, processes, or technologies for sustainable and socially relevant applications.	Level 3: Apply
CO3	Function effectively as an individual and as a leader in diverse teams to comprehend and execute designated tasks.	Level 3: Apply
CO4	Plan and execute tasks utilizing available resources within timelines, following ethical and professional norms.	Level 3: Apply
CO5	Identify technology/research gaps and propose innovative/creative solutions.	Level 4: Analyze
CO6	Organize and communicate technical and scientific findings effectively in written and oral forms.	Level 3: Apply

## CO-PO Mapping

CO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
1	2	2	1	1	-	2	1	-	-	-	-	3
2	3	3	2	3	-	2	1	-	-	-	-	3
3	3	2	-	-	3	-	-	1	-	2	-	3
4	3	-	-	-	2	-	-	1	-	3	-	3
5	3	3	3	3	2	2	-	2	-	3	-	3

## CO-PSO Mapping

CO	PSO 1	PSO 2	PSO 3
1	3	1	2
2	3	2	2
3	2	2	-
4	3	-	3
5	3	-	-