

A DEEP LEARNING APPROACH FOR CLASSIFICATION OF CANCER FROM CT/MRI IMAGES

Submitted in partial fulfillment of the requirements for the award of Bachelor of
Engineering Degree in Electronics and Communication Engineering.

by

Jovie Shinol.J.R (Reg.No: 40130088)

Jenifer.J (Reg.No: 40130087)



**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
SCHOOL OF ELECTRICAL AND ELECTRONICS**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
CATEGORY -1 UNIVERSITY BY UGC**

**Accredited with Grade "A++" by NAAC | 12B Status by UGC | Approved by AICTE
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119**

APRIL – 2024



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

CATEGORY-I UNIVERSITY BY UGC

Accredited "A++" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **JOVIE SHINOL.J.R(40130088), JENIFER.J(40130087)**, who carried out the project entitled **"A DEEP LEARNING APPROACH FOR CLASSIFICATION OF CANCER FROM CT/MRI IMAGES"** under our supervision from December 2023 to April 2024.

Internal Guide

Dr. S. LAKSHMI M.E., Ph.D.,
Professor

Head of the Department

Dr. T. RAVI, M.E., Ph.D.,
Professor

Submitted for Viva voce examination held on_____

Internal Examiner

External Examiner

DECLARATION

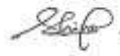
We, **JOVIE SHINOL.J.R(40130088)**, **JENIFER.J(40130087)**, hereby declare that **FINAL YEAR PROJECT REPORT** entitled “**A DEEP LEARNING APPROACH FOR CLASSIFICATION OF CANCER FROM CT/MRI IMAGES**” done by me under the guidance of **Dr.S.Lakshmi M.E.,Ph.D., Professor**, is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Electronics and Communication Engineering.

DATE: 17-04-2024

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATES

1.



2.



ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to The **Board of Management of Sathyabama Institute of Science and Technology** for their kind encouragement in doing this project and for completing it successfully. I am grate full to them.

I convey my thanks to **Dr. N. M. NANDHITHA, M.E., Ph.D., Dean, School of Electrical and Electronics Engineering** and **Dr. T. RAVI, M.E., Ph.D., Head of the Department, Department of Electronics and Communication Engineering** for providing me necessary support during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.S.Lakshmi M.E.,Ph.D., Professor**, Dept. of Electronics and Communication Engineering, for her valuable guidance, suggestions, and constant encouragement for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of Electronics and Communication Engineering who were helpful in many ways for the completion of the project.

ABSTRACT

Advancements in medical imaging have significantly improved cancer diagnosis and treatment planning. However, accurately classifying multiple types of cancer based on CT/MRI images remains a challenging task. In this study, we propose a novel approach to cancer classification by harnessing the power of deep learning models, specifically InceptionV3 and VGG-16 architectures. The objective is to enhance the efficiency and accuracy of cancer diagnosis. Our methodology involves the collection of a diverse and comprehensive dataset of CT/MRI images representing lung, kidney, and brain cancer types. We employ state-of-the-art deep learning architectures, such as convolutional neural networks (CNNs), to extract meaningful features from the medical images. The implications of our research are significant for the medical community, as it provides a powerful tool for accurate cancer classification, aiding oncologists and radiologists in making informed decisions and developing personalized treatment plans for patients with Lung, kidney, and brain cancer. This approach has the potential to improve patient outcomes and contribute to advancements in the field of medical imaging and cancer diagnosis.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	v
	LIST OF FIGURES	viii
	LIST OF TABLES	x
	LIST OF ABBREVIATIONS	xi
1	INTRODUCTION	1
	1.1 FUNDAMENTAL NEURAL NETWORK METHODS	3
2	LITERATURE SURVEY	4
	2.1 INTRODUCTION	4
	2.2 SUMMARY	7
3	EXISTING SYSTEM	8
	3.1 INTRODUCTION	8
	3.2 SUMMARY	9
4	PROPOSED SYSTEM	10
	4.1 ARTIFICIAL INTELLIGENCE	10
	3.1.1 BASIC OPERATIONS OF NEURAL NETWORKS	10
	4.2 IMAGE ACQUISITION	12
	4.2.1 BRAIN DATASET	12
	4.2.2 KIDNEY DATASET	14
	4.2.3 LUNG DATASET	15
	4.3 IMAGE PRE PROCESSING	16
	4.3.1 BRAIN IMAGE PREPROCESSING	16

	4.3.2 KIDNEY IMAGE PREPROCESSING	17
	4.3.3 LUNG IMAGE PREPROCESSING	18
	4.4 TRAINING MODEL	19
	4.4.1 BRAIN MODEL	19
	4.4.2 KIDNEY MODEL	21
	4.4.3 LUNG MODEL	24
	4.5 TESTING MODEL	26
	4.6 FLASK FRAMEWORK	28
	4.7 SYSTEM DESIGN	29
	4.7.1 UML DIAGRAM	29
	4.7.2 USE CASE DIAGRAM	30
	4.7.3 ACTIVITY DIAGRAM	31
	4.7.4 DATA FLOW DIAGRAM	32
5	RESULTS AND DISCUSSIONS	34
	5.1 BRAIN CANCER RESULT AND ANALYSIS	34
	5.2 KIDNEY CANCER RESULT AND ANALYSIS	35
	5.3 LUNG CANCER RESULT AND ANALYSIS	39
	5.4 PREDICTION OUTPUT	42
6	CONCLUSIONS AND FUTURE SCOPE	46
	6.1 CONCLUSIONS	46
	6.2 FUTURE SCOPE	46
	REFERENCES	47

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE NO.
3.1	MODEL ARCHITECTURE FOR ENCODER/DECODER	8
4.1	A MULTI LAYER PERCEPTRON MODEL OF NEURAL NETWORK	11
4.2	ONLINE KAGGLE BRAIN DATASET	13
4.3	KIDNEY CANCER DATASET	14
4.4	LUNG CANCER DATASET	15
4.5	VGG-16 ARCHITECTURE	19
4.6	BLOCK DIAGRAM	21
4.7	INCEPTION V3 ARCHITECTURE	24
4.8	USE CASE DIAGRAM	30
4.9	ACTIVITY DIAGRAM	31
4.10	DATAFLOW DIAGRAM	32
5.1	ACCURACY PLOT	33
5.2	LOSS PLOT	34
5.3	ROC CURVE	34
5.4	PRECISION RECALL ACCURACY LOSS CURVE GRAPH	35
5.5	CONFUSION MATRIX GRAPH	36
5.6	ACCURACY CURVE GRAPH	39
5.7	LOSS CURVE GRAPH	40
5.8	ABSENCE OF BRAIN TUMOR	42

5.9	PRESENCE OF BRAIN TUMOR	43
5.10	NORMAL KIDNEY	43
5.11	CYST IN KIDNEY	44
5.12	NORMAL LUNG	44
5.13	LUNG CANCER	45

LIST OF TABLES

TABLE NO.	TABLE NAME	PAGE NO.
4.1	TRAINED EVALUATION MODEL	26

LIST OF ABBREVIATIONS

CNN	Convolutional Neural Network
MRI	Magnetic Resonance Imaging
CT	Computed Tomography
AI	Artificial Intelligence
BRaTS	Brain Tumor Segmentation
LGG	Low-Grade Glioma
HGG	High-Grade Glioma
EM	Expectation-Maximization
MPM-MAP	Maximum Posterior Marginal - Maximum A Posteriori
UNet	U-Net (a type of convolutional neural network architecture)
ResNet	Residual Network (a type of deep learning architecture)
PCA	Principal Component Analysis
RGB	Red Green Blue
NN	Neural Network
RNN	Recurrent Neural Network
MLP	Multilayer Perceptron
HTML	Hypertext Markup Language
CSS	Cascading Style Sheet

CHAPTER 1

INTRODUCTION

Artificial intelligence (AI) encompasses multiple technologies with the common aim to computationally simulate human intelligence. Deep learning (DL) is a subgroup of ML that focuses on making predictions using multi-layered neural network algorithms inspired by the neurological architecture of the brain. Compared to other ML methods such as logistic regression, the neural network architecture of DL enables the models to scale exponentially with the growing quantity and dimensionality of data. This makes DL particularly useful for solving complex computational problems such as large-scale image classification, natural language processing and speech recognition and translation. Cancer care is undergoing a shift towards precision healthcare enabled by the increasing availability and integration of multiple data types including genomic, transcriptomic and histopathologic data. Moreover, the integration of multiple data types is more resource-intensive than the interpretation of individual data types and needs modelling algorithms that can learn from tremendous numbers of intricate features.

Artificial intelligence (AI) encompasses multiple technologies with the common aim to computationally simulate human intelligence. Machine learning (ML) is a subgroup of AI that focuses on making predictions by identifying patterns in data using mathematical algorithms. Deep learning (DL) is a subgroup of ML that focuses on making predictions using multi-layered neural network algorithms inspired by the neurological architecture of the brain. Compared to other ML methods such as logistic regression, the neural network architecture of DL enables the models to scale exponentially with the growing quantity and dimensionality of data. This makes DL particularly useful for solving complex computational problems such as large-scale image classification, natural language processing and speech recognition and translation.

Cancer care is undergoing a shift towards precision healthcare enabled by the increasing availability and integration of multiple data types including genomic,

transcriptomic and histopathologic data. The use and interpretation of diverse and high-dimensionality data types for translational research or clinical tasks require significant time and expertise. Moreover, the integration of multiple data types is more resource-intensive than the interpretation of individual data types and needs modelling algorithms that can learn from tremendous numbers of intricate features. The use of ML algorithms to automate these tasks and aid cancer detection (identifying the presence of cancer) and diagnosis (characterizing the cancer) has become increasingly prevalent. Excitingly, DL models have the potential to harness this complexity to provide meaningful insights and identify relevant granular features from multiple data types. In this review, we describe the latest applications of deep learning in cancer diagnosis, prognosis and treatment selection. We focus on DL applications for omics and histopathological data, as well as the integration of multiple data types. We provide a brief introduction to emerging DL methods relevant to applications covered in this review. Next, we discuss specific applications of DL in oncology, including cancer origin detection, molecular subtypes identification, prognosis and survivability prediction, histological inference of genomic traits, tumour microenvironment profiling and future applications in spatial transcriptomics, metagenomics and pharmacogenomics. We conclude with an examination of current challenges and potential strategies that would enable DL to be routinely applied in clinical settings.

Covering all DL methods in detail is outside the scope of this review; rather, we provide a high-level summary of emerging DL methods in oncology. DL utilises artificial neural networks to extract non-linear, entangled and representative features from massive and high-dimensional data. A deep neural network is typically constructed of millions of densely interconnected computing neurons organized into consecutive layers. Within each layer, a neuron is connected to other neurons in the layer before it, from which it receives data, and other neurons in the layer after it, to which it sends data. When presented with data, a neural network feeds each training sample, with known ground truth, to its input layer before passing the information down to all succeeding layers (usually called hidden layers). This information is then multiplied, divided, added and subtracted millions of times before it reaches the output layer, which

becomes the prediction. For supervised deep learning, each pair of training sample and label is fed through a neural network while its weights and thresholds are being adjusted to get the prediction closer to the provided label. When faced with unseen (test) data, these trained weights and thresholds are frozen and used to make predictions.

1.1 FUNDAMENTAL NEURAL NETWORK METHODS

There are multiple neural network-based methods, all with different advantages and applications. Multilayer perceptron (MLP), recurrent neural network (RNN) and convolutional neural network (CNN) are the most fundamental and are frequently used as building blocks for more advanced techniques. MLPs are the simplest type of neural networks, where neurons are organized in consecutive layers so that signals travel through the network in one direction (from input to output). Although MLPs can perform well for generic predictions, they are also prone to overfitting. RNNs process an input sequence one element at a time, while maintaining history of all past elements in hidden 'state vector(s)'. Output predictions are made at every element using information from the current element and also previous elements. RNNs are typically used for analyzing sequential data such as text, speech or DNA sequences. By contrast, CNNs are designed to draw spatial relationships from image data. CNNs traverse an image and apply small feature-filter matrices, i.e. convolution filters, to extract granular features. Features extracted by the last convolution layer are then used for making predictions. CNNs have also been adapted for analysis of non-image data, e.g. genomic data represented in a vector, matrix or tensor format. A review by Dias and Torkamani described in detail how MLPs, RNNs and CNNs operate on biomedical and genomics data. Moreover, the use of MLPs, RNNs and CNNs to assist clinicians and researchers has been proposed across multiple oncology areas, including radiotherapy, digital histopathology and clinical and genomic diagnostics. While routine clinical use is still limited, some of the models have already been FDA-approved and adopted into a clinical setting, for example CNNs for the prediction of malignancy in pulmonary nodules detected by CT, and prostate and cancer diagnosis prediction using digital histopathology.

CHAPTER 2

LITERATURE SURVEY

2.1 INTRODUCTION

This literature review seeks to provide a comprehensive overview of the current state of research and development in the field of multicancer detection and treatment. By synthesizing findings from a diverse range of studies, we aim to elucidate the key methodologies, technologies, and clinical implications associated with multicancer strategies. Additionally, this review will explore the challenges, limitations, and future directions in advancing the implementation of multicancer approaches in clinical practice.

Pan. S.J and Yang .Q (2020) survey focused on categorizing and reviewing the current progress on transfer learning for classification, regression and clustering problems. In this survey, they discussed the relationship between transfer learning and other related machine learning techniques such as domain adaptation, multitask learning and sample selection bias, as well as co-variate shift. They also explored some potential future issues in transfer learning research. In this survey article, they reviewed several current trends of transfer learning.

A Survey on Brain Tumor Detection Using Image Processing Techniques by Luxit Kapoor .S and Sanjeev .S (2018). This paper surveys the various techniques that are part of Medical Image Processing and are prominently used in discovering brain tumors from MRI Images. Based on that research this Paper was written listing the various techniques in use. A brief description of each technique is also provided. Also, of All the various steps involved in the process of detecting Tumors, Segmentation is the most significant.

Banerjee . S et al.,(2016), reports the accuracy achieved by seven standard classifiers, viz. i) Adaptive Neuro- Fuzzy Classifier (ANFC), ii) Naive Bayes(NB), iii) Logistic Regression (LR), iv) Multilayer Perceptron(MLP), v) Support Vector Machine

(SVM), vi) Classification and Regression Tree (CART), and vii) k-nearest neighbors (k-NN). The accuracy reported is on the BRaTS 2015 dataset (a subset of BRaTS 2017 dataset) which consists of 200 HGG and 54 LGG cases. 56 three-dimensional quantitative MRI features extracted manually from each patient MRI and used for the classification.

Moradi .M et al., (2021), compared different techniques to differentiate lung cancer nodules from non-nodules. To reduce/eliminate the false positive predictions they have come up with 3D Convolutional Neural Network Technique. Nodules exist in different sizes and using just one CNN can result in false detections. So, they divided the nodules into four groups according to their size. And they have used four different sizes of 3D CNN. They combined all those 4 classifiers to get better results. Each CNN consists of a number of 3D CNN which are all varying sizes. All 4 classifiers were combined in order to produce results which were better. A combination of Max pooling layer and convolutional layer were used to produce each CNN. The activation function used here is ReLU. SoftMax layer accompanied by a fully connected layer is used to produce the output finally. Nodules size varies from 3mm to 3cm so by using just one layer, the prediction could be wrong for either very small nodules or very large values. So, they fused all the 4 CNNs and sent their output values (predicted values) to a final classifier. They have chosen a logistic regression classifier that takes inputs from 4 CNNs and produces a final prediction. They have implemented logistic regression by using a decision tree classifier and gradient boosting model. LUNA16 dataset was used in this to train the complete model. LUNA16 is based on the CT images of the LIDC dataset. As a result, they saw that the result by the fused classifier is better than each of the solo classifiers.

Bohdan Chapliuk .G et al.(2018) [4] applied neural networks C3D and 3D DenseNet to detect lung cancer using CT images. These Neural networks were applied to whole lung 3D images and two-stage approaches (for segmentation and classification, two different neural networks are trained.) and further compared.

Hunnur S. S et al., (2017), the authors proposed a method to detect brain tumors mainly based on Thresholding approach and morphological operations. It also calculates the area of the tumor and displays the stage of the tumor. MRI images of the brain are used as input and the images are converted into grayscale images. High pass filter is used to remove the noise present in the converted images. The median filter is applied to remove the impulse noise. Thresholding is used to extract the object from the background by selecting a threshold value. Morphological operations- dilation and erosion are done. Tumor region is detected and then the image is shrunk to remove the unwanted details present in the images. The tumor area is calculated and at last, the stage of the tumor patient is displayed.

Qinghai zhang .A et al., (2021), [3] proposed a method for designing of lung nodule detection system which is automatic. The dataset used for the proposed method is LIDC-IRDI public dataset. The proposed method used for this study is Multi-Scene Deep Learning Framework which contains several steps. CT images are given as input and the probability distribution of distinct gray levels is obtained by threshold segmentation that is Histogram. Correcting the smooth lung outlines is the main aim for the lung parenchyma segmentation process. The replacement of the vein system in the lung helps to identify the nodule structure. Vessel filters are used for removing the vessels which reduce the number of false positive. The design of CNN contains a pooling layer, a convolutional layer, and a fully integrated layer. Segmentation and classification identify Class 1 and Class2 that are two class of image data and discrete images which are separated from the lung images respectively [31]. Segmentation is done to identify cancerous tumor cells in lungs. The accuracy of the determined nodules is determined by four different types of CNN architecture.

Marroquin J.L et al., (2002), presented the automated 3d segmentation for brain MRI scans. Using a separate parametric model in preference to a single multiplicative magnificence will lessen the impact on the intensities of a grandeur. Brain atlas is hired to find nonrigid conversion to map the usual brain. This transformation is further used to segment the brain from nonbrain tissues, computing prior probabilities and finding

automatic initialization and finally applying the MPM-MAP algorithm to find out optimal segmentation. Major findings from the study show that the MPM-MAP algorithm is comparatively robust than EM in terms of errors while estimating the posterior marginal. For optimal segmentation, the MPM-MAP algorithm involves only the solution of linear systems and is therefore computationally efficient. Astinaminz .J et al.(2002) implemented an operative automatic classification approach for brain image that projected the usage of the AdaBoost gadget mastering algorithm. The proposed system includes three main segments. Pre-processing has eradicated noises in the datasets and converted images into grayscale. Median filtering and thresholding segmentation are implemented in the pre-processed image.

In their work titled "Efficient Breast Cancer Detection via Cascade Deep Learning Network," Bitu Asadi and Qurban Memon leverage UNet and ResNet architectures for segmentation and classification. Despite achieving high accuracy, the study highlights a common concern in medical deep learning—interpretability. The review underscores the importance of transparent models in medical AI, explores techniques for improving interpretability, showcases case studies, and outlines challenges, guiding future research to strike a balance between performance and interpretability.

2.2 SUMMARY

The literature review encompasses a wide array of studies focusing on various techniques and methodologies for the detection and treatment of multicancer cases. It begins by providing a comprehensive overview of the current state of research and development in this rapidly evolving field. The review synthesizes findings from diverse studies, elucidating key methodologies, technologies, and clinical implications associated with multicancer strategies. This collection of studies covers a range of innovative approaches, including advanced imaging techniques, machine learning algorithms, and segmentation methods, all contributing to the evolving landscape of multicancer research and care.

CHAPTER 3

EXISTING SYSTEM

3.1 INTRODUCTION

In the domain of cancer detection and classification, various advanced methodologies have been employed to enhance accuracy and reliability. One such technique is the Autoencoder, a type of artificial neural network known for its unsupervised learning capability. Autoencoders are adept at learning efficient representations of input data by minimizing reconstruction errors. They have found application in compressing and reconstructing high-dimensional data, particularly in medical imaging, where preserving critical features is paramount.

Another notable method is the Least Squares Support Vector Machine (LS-SVM) classifier. Unlike conventional SVMs, LS-SVM focuses on minimizing the regression error, making it well-suited for regression tasks and classification with continuous outputs. This method is valuable in scenarios where linear separation boundaries are not optimal, providing robustness in complex, real-world datasets.

In the realm of signal processing, both Stationary Wavelet Transforms (SWT) and Discrete Wavelet Transform (DWT) have played a pivotal role in extracting relevant features from signals. SWT is adept at capturing both frequency and time-domain characteristics, making it particularly useful in analyzing non-stationary signals. DWT, on the other hand, excels in decomposing signals into multiple scales, allowing for a multi-resolution analysis that can unveil subtle variations in data.

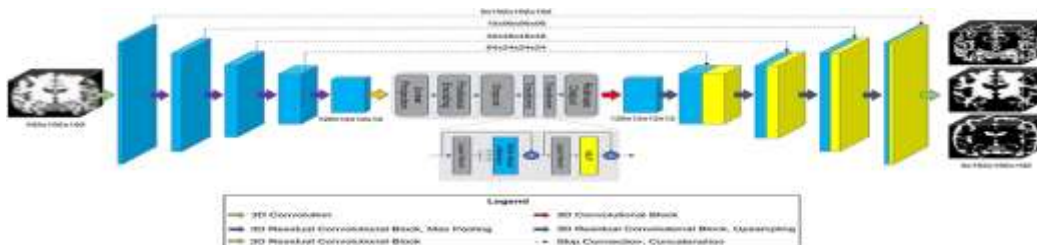


Fig.3.1: Model architecture for encoder/decoder

Furthermore, the SVM-based segmentation method leverages the power of Support Vector Machines for precisely delineating regions of interest in medical images. SVMs, known for their versatility in handling both linear and non-linear decision boundaries, are well-suited for tasks that demand high precision and accuracy, such as image segmentation in cancer detection is shown in the Fig.3.1.

3.2 SUMMARY

In cancer detection, advanced methods like Autoencoders for data representation and LS-SVM for regression tasks play key roles. Wavelet transforms (SWT and DWT) extract features from signals effectively. SVM-based segmentation ensures precise image delineation. These methods collectively enhance accuracy in cancer diagnosis.

CHAPTER 4

PROPOSED SYSTEM

4.1 ARTIFICIAL INTELLIGENCE:

Artificial intelligence (AI) is the simulation of human intelligence processes by machines, especially computer systems enabling it to even mimic human behavior. Its applications lie in fields of Computer Vision, Natural Language Processing, Robotics, Speech Recognition, etc. Advantages of using AI are improved customer experience, accelerate speed to market, develop sophisticated products, enable cost optimization, enhance employee productivity and improve operational efficiency. Machine Learning (ML) is a subset of AI which is programmed to think on its own, perform social interaction, learn new information from the provided data and adapt as well as improve with experience. Although training time via Deep Learning (DL) methods is more than Machine Learning methods, it is compensated by higher accuracy in the former case. Also, DL being automatic, large domain knowledge is not required for obtaining desired results unlike in ML.

4.1.1 BASIC OPERATIONS OF NEURAL NETWORKS:

Neural Networks (NN) form the base of deep learning, a subfield of machine learning where the algorithms are inspired by the structure of the human brain. NN take in data, train themselves to recognize the patterns in this data and then predict the outputs for a new set of similar data. NN are made up of layers of neurons. These neurons are the core processing units of the network. First, we have the input layer which receives the input; the output layer predicts our final output. In between, exist the hidden layers which perform most of the computations required by our network.

Our brain tumor images are composed of 128 by 128 pixels which make up for 16,384 pixels. Each pixel is fed as input to each neuron of the first layer. Neurons of one layer are connected to neurons of the next layer through channels. Each of these channels is assigned a numerical value known as 'weight '. The inputs are multiplied to

the corresponding weight and their sum is sent as input to the neurons in the hidden layer. Each of these neurons is associated with a numerical value called the bias which is then added to the input sum. This value is then passed through a threshold function called the 'activation function '. The result of the activation function determines if the particular neuron will get activated or not. An activated neuron transmits data to the neurons of the next layer over the channels. In this manner the data is propagated through the network this is called' forward propagation '. In the output layer the neuron with the highest value fires and determines the output. The values are basically a probable.

The predicted output is compared against the actual output to realize the error in prediction. The magnitude of the error gives an indication of the direction and magnitude of change to reduce the error. This information is then transferred backward through our network. This is known as 'back propagation '. Now based on this information the weights are adjusted. This cycle of forward propagation and back propagation is iteratively performed with multiple inputs. This process continues until our weights are assigned such that the network can predict the type of tumor correctly in most of the cases. This brings our training process to an end. NN may take hours or even months to train but time is a reasonable trade-off when compared to its scope. Several experiments show that after pre-processing MRI images, neural network classification algorithm was the best more specifically CNN(Convolutional Neural Network) as compared to Support Vector Machine(SVM),Random Forest Field.

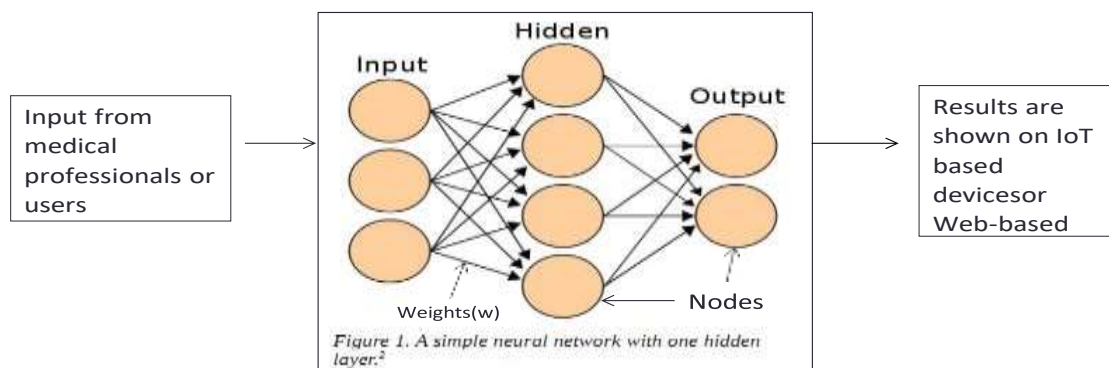


Fig: 4.1: A Multi-Layer perceptron Model of Neural Network

A multi-layer perceptron neural network model, showcasing the interconnected layers of nodes. Each node represents a neuron, with connections indicating information flow, ultimately enabling complex computations for tasks like classification and regression is shown in Fig 4.1.

4.2 IMAGE ACQUISITION:

Images can be in the form of .csv (comma separated values), .dat (data) files in grayscale, RGB, or HSV or simply in .zip file as was in the case of our online Kaggle dataset. It contained 98 healthy MRI images and 155 tumor infected MRI images.

4.2.1 Brain dataset:

The Multimodal Brain Tumor Segmentation (BRaTS) MICCAI has always been focusing on the evaluation of state-of-the-art methods for the segmentation of brain tumors in magnetic resonance imaging (MRI) scans. Ample multi-institutional routine clinically-acquired multimodal MRI scans of glioblastoma (GBM) and lower grade glioma (LGG), with pathologically confirmed diagnosis and available OS, was provided as the training, validation and testing data for BRaTS 2015 challenge. All BRaTS multimodal scans are available as NIfTI files (.nii.gz) and these multimodal scans describe a) native (T1) and b) post-contrast T1- weighted (T1c), c) T2-weighted (T2), and d) T2 Fluid Attenuated Inversion Recovery (FLAIR) volumes, and were acquired with different clinical protocols and various scanners from multiple institutions. They described a mixture of pre- and post-operative scans and their ground truth labels have been annotated by the fusion of segmentation results from algorithms. All the imaging datasets have been segmented manually, by one to four raters, following the same annotation protocol, and their annotations were approved by experienced neuro-radiologists. Annotations comprise the whole tumor, the tumor core (including cystic areas), and the C- enhancing tumor core.

The dataset contains 2 folders for the purpose of training and testing. The 'train' folder contains 2 sub-folders of HGG and LGG cases-220 patients of HGG and 27 patients of LGG. The test folder contains brain images of 110 Patients with HGG and

LGG cases combined. There are 5 different MRI image modalities for each patient which are T1, T2, T1C, FLAIR, and OT (Ground truth of tumor Segmentation). All these image files are stored in mha format and are of the size of 240x240, resolution of (1 mm³) and skull- stripped. In the ground truth images, each voxel is labelled with zeros and non-zeros, corresponding to the normal pixel and parts of tumor cells, respectively.

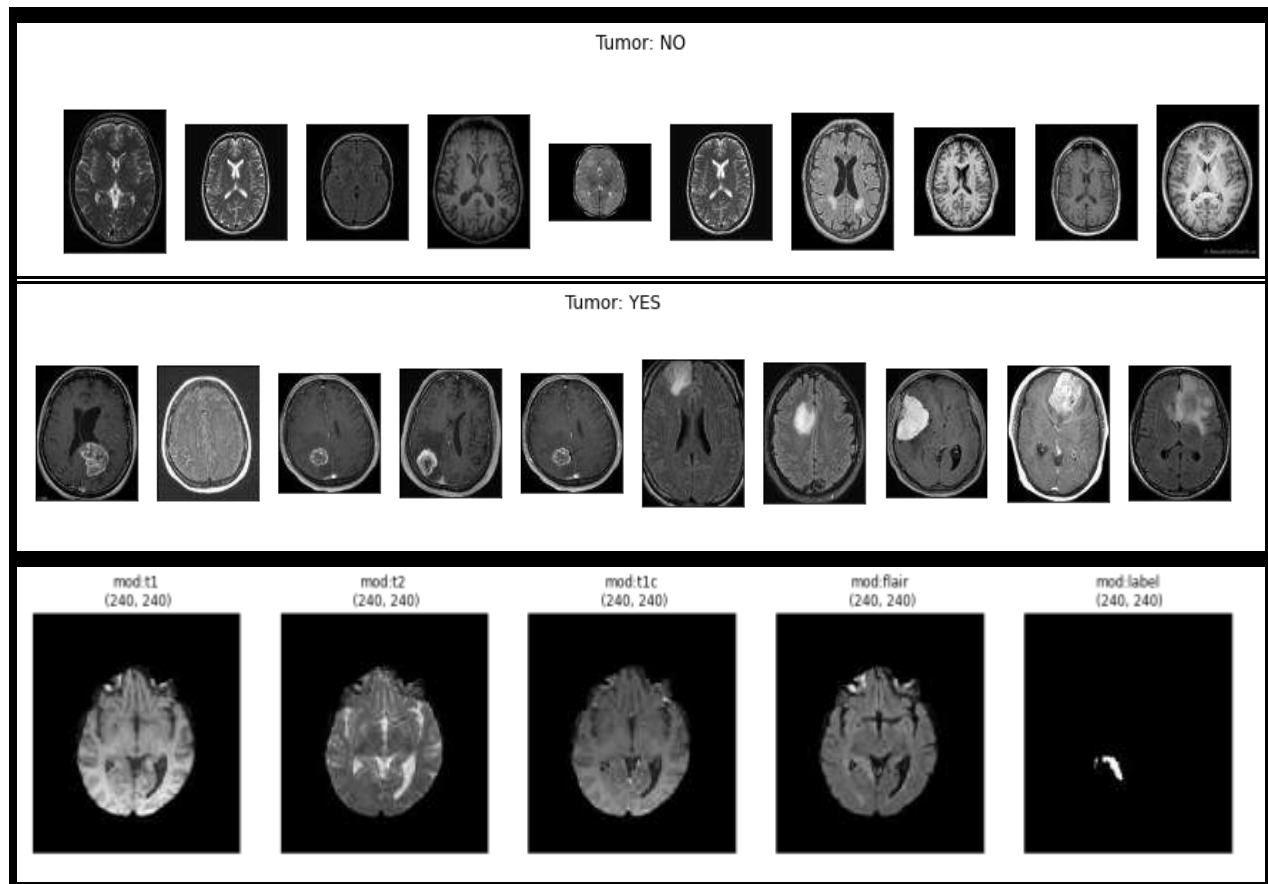


Fig: 4.2: Online Kaggle Brain Cancer Dataset

The online Kaggle Brain cancer dataset, a digital repository containing labeled medical images of brain scans. This dataset is a valuable resource for training and

evaluating machine learning models for brain cancer detection and classification tasks is shown in Fig.4.2.

4.2.2 Kidney Dataset:

The dataset was collected from PACS (Picture archiving and communication system) from different hospitals in Dhaka, Bangladesh where patients were already diagnosed with having a kidney tumor, cyst, normal or stone findings.

Both the Coronal and Axial cuts were selected from both contrast and non-contrast studies with protocol for the whole abdomen and urogram.

The Dicom study was then carefully selected, one diagnosis at a time, and from those we created a batch of Dicom images of the region of interest for each radiological finding.

Following that, we excluded each patient's information and meta data from the Dicom images and converted the Dicom images to a lossless jpg image format. After the conversion, each image finding was again verified by a radiologist and a medical technologist to reconfirm the correctness of the data. The dataset contains 12,446 unique data within it in which the cyst contains 3,709, normal 5,077, stone 1,377 and tumor 2,283.

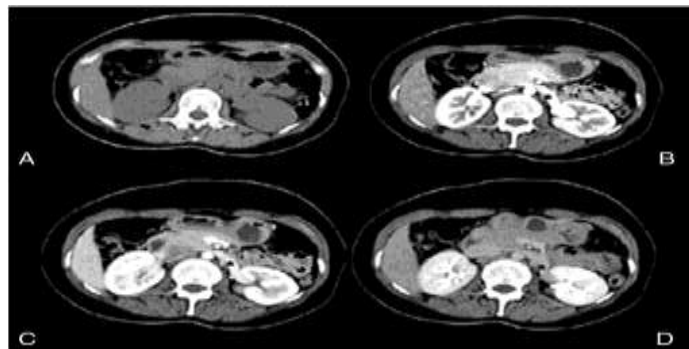


Fig: 4.3: Kidney Cancer Dataset

The kidney cancer dataset, likely a visual depiction or graphical representation of a collection of medical data related to kidney cancer. This dataset is essential for

conducting research and developing algorithms aimed at improving the diagnosis and treatment of kidney cancer is shown in Fig 4.3.

4.2.3 Lung Cancer dataset:



Fig: 4.4: Lung Cancer Dataset

The Graphical overview of the Lung Cancer dataset comprises structured information related to lung cancer cases, essential for conducting research, analysis, and the development of models for lung cancer detection is shown in Fig 4.4.

1. Adenocarcinoma: Adenocarcinoma is a type of lung cancer that begins in the cells of the glands that line the airways. It is one of the most common types of lung cancer and tends to occur in the outer regions of the lungs. Adenocarcinoma is often associated with non-smokers.

2. **Large Cell Carcinoma:** Large cell carcinoma is another type of non-small cell lung cancer. It is called "large cell" because the cancer cells appear large and abnormal under a microscope. Large cell carcinoma accounts for a smaller percentage of lung cancers.
3. **Squamous Cell Carcinoma:** Squamous cell carcinoma is a type of non-small cell lung cancer that usually develops in the lining of the bronchial tubes. It is often linked to a history of smoking and is one of the subtypes of non-small cell lung cancer.
4. **Normal Lung Images:** This refers to images of healthy lung tissues without any signs of cancer or abnormalities. These images serve as a comparison or baseline for understanding and diagnosing lung diseases.

Data Augmentation:

Data augmentation consists of Grey Scaling(RGB/BW to ranges of grey),Reflection(vertical/horizontal flip),Gaussian Blur(reduces image noise),Histogram equalization(increases global contrast),Rotation(may not preserve image size),Translation(moving the image along x or y axis), linear transformation such as random rotation (0-10 degrees), horizontal and vertical shifts, and horizontal and vertical flips. Data augmentation is done to teach the network desired invariance and robustness properties, when only few training samples are available.

4.3 IMAGE PRE-PROCESSING:

In the first step of pre-processing, the memory space of the image is reduced by scaling the gray-level of the pixels in the range 0-255.

4.3.1 Brain Image Preprocessing

1. Image Rescaling:

- **Objective:** Scaling pixel values to a range between 0 and 1.

- **Reasoning:** Neural networks generally perform better when input features are on a similar scale. Rescaling ensures that pixel values, which range from 0 to 255, are normalized to a range suitable for training neural networks.

2. Data Augmentation:

- **Objective:** Generate variations of the training images through horizontal and vertical flips.
- **Reasoning:** Data augmentation helps increase the diversity of the training set, reducing the risk of overfitting. Flipping the images horizontally and vertically provides additional training samples without the need for collecting more data. This aids in improving the model's ability to handle variations in tumor images. `rescale=1./255`: Normalize pixel values to the range [0, 1].
- `horizontal_flip=2`: Randomly flip images horizontally. The value 2 indicates that the horizontal flip is applied randomly with a probability of 50% for each image.
- `vertical_flip=2`: Randomly flip images vertically. Similar to horizontal flip, the value 2 indicates a 50% probability of vertical flip for each image.
- `validation_split=0.1`: Specifies the fraction of the data to be used for validation. In this case, 10% of the data is reserved for validation during training.

4.3.2 Kidney Image Preprocessing

1. Dataset Splitting:

- **Objective:** Split the original dataset into training, validation, and test sets.

- Reasoning: This step ensures that the model is trained on a diverse dataset, validated on a separate subset to tune hyperparameters, and finally evaluated on an independent test set to assess generalization.

2. Image Rescaling:

- Objective: Normalize pixel values to the range [0, 1].
- Reasoning: Rescaling ensures that pixel values are on a consistent scale, making it easier for the neural network to learn and converge during training.

3. Image Data Generators:

- Objective: Create data generators for training, validation, and test sets.
- Reasoning: Data generators enable efficient loading and preprocessing of images in batches, reducing memory usage and allowing the model to process large datasets.

4.3.3 Lung Image Preprocessing

Training Data Generator:

- Objective: Generate augmented batches of training images.
- Steps:
 - rescale=1.0/255.0: Normalize pixel values to the range [0, 1].
 - horizontal_flip=True: Randomly flip images horizontally for increased variation.
 - fill_mode='nearest': Fill points outside the input boundary using the nearest available pixel.
 - zoom_range=0.2: Randomly zoom into images by a factor of up to 20%.

- shear_range=0.2: Apply shear transformations with a maximum shear intensity of 20%.
- width_shift_range=0.2 and height_shift_range=0.2: Randomly shift the width and height of images by a maximum of 20%.
- rotation_range=0.4: Randomly rotate images by up to 40 degrees.

4.4 TRAINING MODEL

4.4.1 Brain Model:

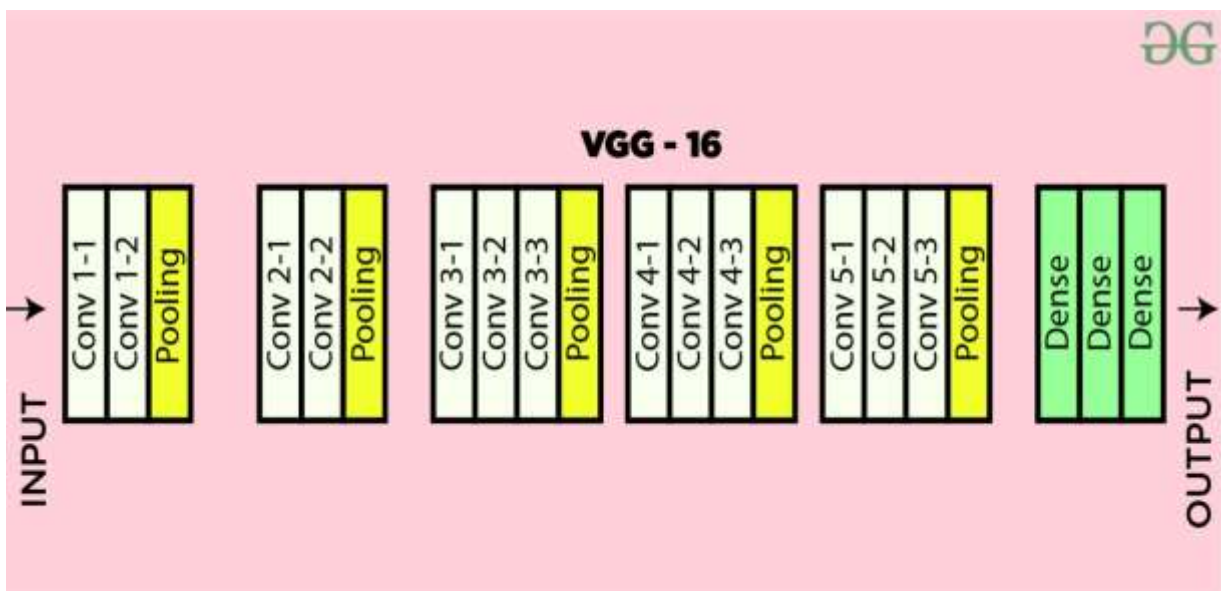


Fig: 4.5: Vgg-16 Architecture

The VGG-16 model is a convolutional neural network(CNN) architecture was proposed by the Visual Geometry Group (VGG) at the University of Oxford. It is characterized by its depth, consisting of 16 layers including 13 convolutional layers and 3 fully connected layers as shown in Fig 4.5.

1. Architecture:

- VGG16 consists of 16 convolutional and fully connected layers.

- The architecture is uniform, with only 3x3 convolutional filters and 2x2 max-pooling layers used throughout.
- Stacking multiple small filters allows the network to learn more complex features while keeping the number of parameters manageable.

2. Layers:

- The network comprises 13 convolutional layers and 3 fully connected layers.
- Convolutional layers are denoted as ConvX-Y, where X is the block number, and Y is the layer number within the block.
- The fully connected layers are followed by a softmax layer for classification.

3. Filter Size and Stride:

- All convolutional layers use 3x3 filters, and all pooling layers use 2x2 filters.
- The stride for convolutional layers is set to 1, and the stride for pooling layers is set to 2.

4. Pooling Layers:

- VGG16 uses max-pooling layers after each block of convolutional layers.
- The pooling layers down-sample the spatial dimensions, reducing the computational load and providing a degree of translation invariance.

5. Fully Connected Layers:

- The three fully connected layers have 4096 neurons each.
- The final layer has the number of neurons equal to the number of classes in the specific classification task (e.g., 1000 for ImageNet).

6. Activation Function:

- Rectified Linear Units (ReLU) are used as the activation function after each convolutional and fully connected layer.

7. Normalization:

- Batch normalization was not included in the original VGG16 architecture. It has been introduced in later architectures to improve training stability and speed.

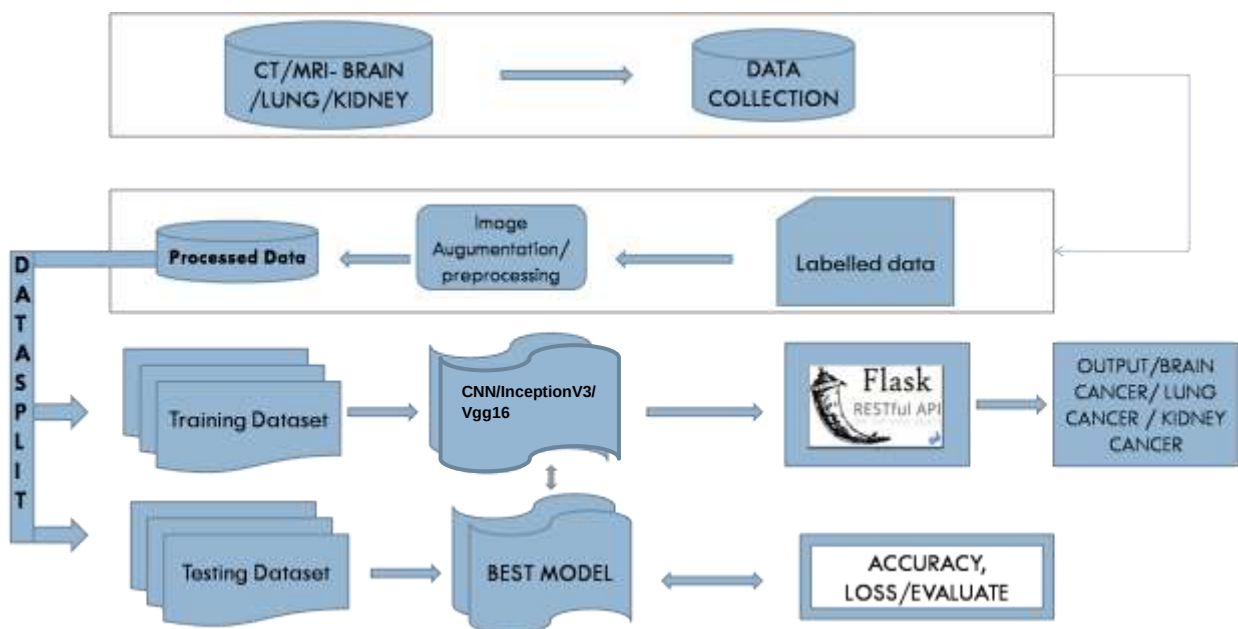


Fig: 4.6: Block Diagram

The training and testing will be done and accuracy will be evaluates based on the block diagram as a shown in Fig 4.6.

4.4.2 Kidney model:

Convolutional Neural Networks (CNNs) are a class of deep neural networks designed for processing structured grid data, such as images. CNNs use a specialized architecture that takes advantage of the spatial structure in the input data. Let's break down the main types of layers typically used in a CNN:

1. Convolutional Layer (Conv2D):

- **Functionality:** The primary building block of a CNN. Convolution involves passing a filter (also called a kernel) over the input image to extract features. Each filter learns to recognize different patterns or features in the input.
- **Parameters:**
 - **Number of filters:** Determines the depth of the output volume.
 - **Filter size (kernel size):** Specifies the spatial dimensions of the filter.
 - **Activation function:** Introduces non-linearity after convolution.

2. Activation Layer (ReLU - Rectified Linear Unit):

- **Functionality:** Applied element-wise after convolution to introduce non-linearity. ReLU is a common activation function that replaces all negative pixel values in the feature map with zero.
- **Purpose:** Enhances the network's ability to learn complex patterns and relationships.

3. Pooling Layer (MaxPooling2D or AveragePooling2D):

- **Functionality:** Reduces the spatial dimensions (width and height) of the input volume by down-sampling. Max pooling takes the maximum value from a group of neighboring pixels, while average pooling takes the average.
- **Purpose:** Helps reduce computational complexity, makes the network more robust to variations in position, and reduces overfitting.

4. Flatten Layer:

- **Functionality:** Converts the multi-dimensional output of the previous layer into a one-dimensional array. It flattens the spatial structure, preparing the data for fully connected layers.

- Purpose: Transitions from the convolutional/pooling layers to the fully connected layers.

5. Fully Connected Layer (Dense):

- Functionality: Traditional neural network layer where each neuron is connected to every neuron in the previous and next layers.
- Parameters:
 - Number of neurons: Determines the depth of the output volume.
 - Activation function: Introduces non-linearity.
- Purpose: Captures complex relationships and dependencies in the data, providing high-level abstractions.

6. Output Layer:

- Functionality: Produces the final output of the network.
- Parameters:
 - Number of neurons: Typically corresponds to the number of classes in classification tasks.
 - Activation function: Often softmax for multi-class classification to convert raw scores into probabilities.
- Purpose: Generates predictions for the task at hand.

4.4.3 Lung Model:

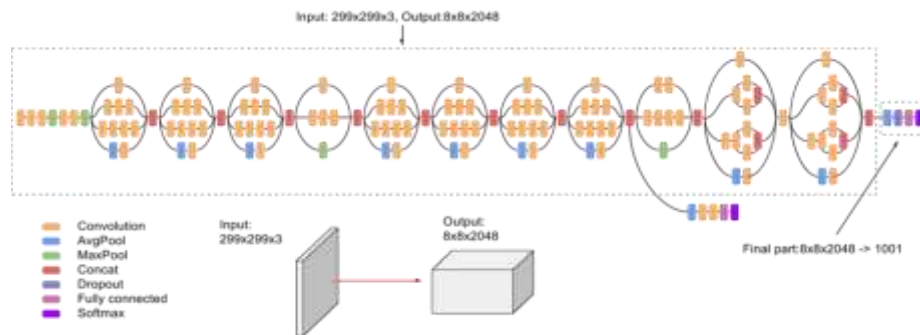


Fig: 4.7: Inception V3 Architecture

Inception-v3 is a convolutional neural network architecture from the Inception family that makes several improvements including using Label Smoothing, Factorized 7x 7 convolutions , and the use of auxiliary classifier to propagate label information lower down the network as shown n Fig 4.7

1. Architecture:

- InceptionV3 is a deep neural network architecture with 48 layers.
- It introduces the concept of "Inception modules," which consist of multiple parallel convolutional branches with different filter sizes.
- The use of these parallel branches allows the network to capture both fine and coarse-grained features.

2. Inception Modules:

- The main building block is the Inception module, which is designed to capture features at different spatial scales.
- Each Inception module typically includes 1x1, 3x3, and 5x5 convolutions, as well as a pooling operation.

- The outputs of these operations are concatenated along the depth dimension.

3. Factorization:

- InceptionV3 uses factorization techniques to reduce the number of parameters.
- It employs 1x1 convolutions for dimensionality reduction before applying larger convolutions (e.g., 3x3 or 5x5).
- This helps in making the network more computationally efficient.

4. Auxiliary Classifiers:

- InceptionV3 includes auxiliary classifiers at intermediate layers during training.
- These auxiliary classifiers are used to combat the vanishing gradient problem and provide additional supervision during training.

5. Batch Normalization:

- Batch normalization is applied after each convolutional and fully connected layer.
- It helps stabilize and accelerate training by normalizing the inputs to the network.

6. Global Average Pooling (GAP):

- Instead of fully connected layers at the end, InceptionV3 uses global average pooling.
- Global average pooling spatially averages the feature maps, resulting in a fixed-size output.

4.5 TESTING MODEL

- True Positive: You predicted positive, and it's true.
- True Negative: You predicted negative, and it's true.
- False Positive: (Type 1 Error): You predicted positive, and it's false.
- False Negative: (Type 2 Error): You predicted negative, and it's false.

Table 4.1: Trained Machine Learning Model

Confusion Matrix		Target			
		Positive	Negative		
Model	Positive	a	b	Positive Predictive Value	$a/(a+b)$
	Negative	c	d	Negative Predictive Value	$d/(c+d)$
		Sensitivity	Specificity	Accuracy = $(a+d)/(a+b+c+d)$	
		$a/(a+c)$	$d/(b+d)$		

This module we test the trained machine learning model using the test dataset is shown in Table 4.1.

Accuracy:

- The most commonly used metric to judge a model and is actually not a clear indicator of the performance. The worse happens when classes are imbalanced.

$$\frac{TP + TN}{TP + FP + TN + FN} \dots \dots \dots (4.1)$$

Precision:

- Percentage of positive instances out of the total predicted positive instances.

$$\frac{TP}{TP + FP} \dots \dots \dots (4.2)$$

Recall/Sensitivity/True Positive Rate:

- Percentage of positive instances out of the total actual positive instances.

$$\frac{TP}{TP + FN} \dots \dots \dots (4.3)$$

F1 score

- It is the harmonic mean of precision and recall. This takes the contribution of both, so higher the F1 score, the better.

$$\frac{2}{\frac{1}{precision} + \frac{1}{recall}} = \frac{2 * precision * recall}{precision + recall} \dots \dots \dots (4.4)$$

PREDICTION

Flask Integration:

- A Flask web application is created to provide a user interface for making predictions using the trained models.
- Flask routes are defined for handling different functionalities, including loading the models and making predictions.
- HTML templates are used for creating user interfaces to take input features and display prediction results.

User Interaction:

- Users interact with the Flask web application by filling in a form with input features (e.g., gestational age, mother's age, smoking status) to predict birth weight.
- The submitted form data is sent to the server, and predictions are generated using the trained models.
- Prediction results are then displayed to the user.

Flask is a lightweight web framework for Python that allows you to build web applications quickly and easily. Flask works by Model-View-Controller (MVC) architecture pattern, where the model represents the data, the view represents the user interface, and the controller acts as an intermediary between the model and the view.

4.6 Flask Framework

The framework is the basis upon which software programs are built. It serves as a foundation for software developers, allowing them to create a variety of applications for certain platforms. It is a set of functions and predefined classes used to connect with the system software and handle inputs and outputs.

It simplifies the life of a developer while giving them the ability to use certain extensions and makes the online applications scalable and maintainable.

Frontend Development vs Backend Development

1.The front end of a website is the area with which the user immediately interacts. It contains everything that users see and interact with: text colours and styles, images and videos, graphs and tables, the navigation menu, buttons, and colours. HTML, CSS, and JavaScript are used in developing the front end.

2.The backend of a website refers to the server-side of the website. It saves and organizes data and ensures that everything on the client-side of the website functions properly. It is the section of the website that you are unable to interact with or access. It is the part of the software that has no direct communication with the users. Back End development is done using Java, Python, C#, and JavaScript.

Flask is used for developing web applications using python, implemented on Werkzeug and Jinja2. Advantages of using Flask framework are:

- There is a built-in development server and a fast debugger provided.
- Lightweight.

- Secure cookies are supported.
- Templating using Jinja2.
- Request dispatching using REST.
- Support for unit testing is built-in.

4.7 SYSTEM DESIGN

4.7.1 UML DIAGRAM

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML. The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems. The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

GOALS:

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.

2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.

4.7.2 USE CASE DIAGRAM:

A Use case Diagram is used to present a graphical overview of the functionality provided by a system in terms of actors, their goals and any dependencies between those use cases. Use case diagram consists of two parts:

Use case: A use case describes a sequence of actions that provided something of measurable value to an actor and is drawn as a horizontal ellipse.

Actor: An actor is a person, organization or external system that plays a role in one or more interaction with the system.

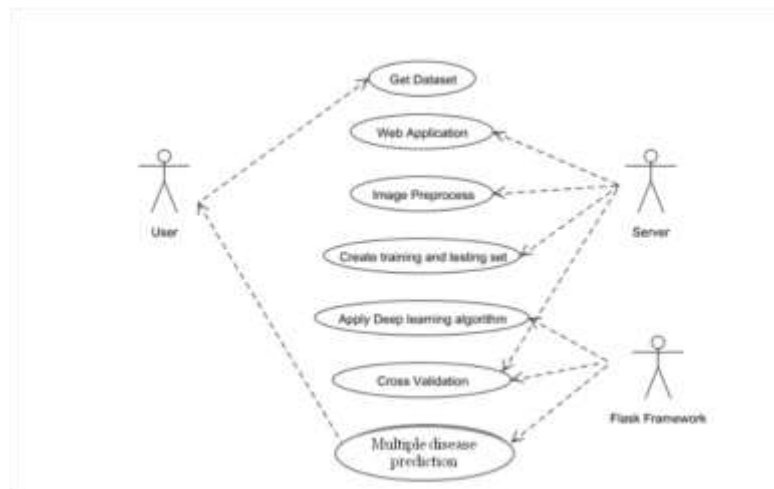


Fig: 4.8: Use Case Diagram

As shown in Fig 4.8, This use case illustrates how deep learning models for multiple disease prediction can be integrated into routine health checkups, enhancing the quality of healthcare assessments and enabling personalized, preventive healthcare. It empowers both patients and healthcare professionals with valuable insights and actionable information to improve overall health and well-being.

4.7.3 ACTIVITY DIAGRAM:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system.

Initial Node: The initial node represents the starting point of the process or system.

Activities: Activities represent the tasks or actions that are performed in the system or process. Each activity is represented by a rectangle with rounded corners.

Decision Points:

Decision points are represented by a diamond-shaped box and indicate a decision point in the system or process. The decision point may have multiple possible outcomes.

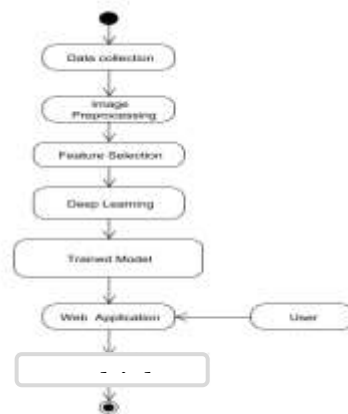


Fig: 4.9: Activity Diagram

As shown in Fig 4.9, The "activity" in this context is a comprehensive process involving data collection, analysis, modeling, and decision-making to predict, diagnose, and manage multiple diseases simultaneously. These activities are crucial in the field of healthcare to improve disease detection, provide more effective patient care, and advance medical research

4.7.4 DATA FLOW DIAGRAM:

A data flow diagram (DFD) is a graphical representation of the flow of data through a system or process. It consists of different components that show how data moves through the system. Here are the main components of a data flow diagram:

Entities:An entity is an external agent that interacts with the system. This can be a person, organization, or other system that sends or receives data.

Processes:A process is an activity or task that transforms data into information. It can include functions such as calculations, validations, and data manipulations.

Data Flows: Data flows represent the movement of data between entities, processes, and storage locations. They are represented by arrows that show the direction of data flow.

Data Stores:Data stores are locations where data is stored within the system. This can include databases, files, and other storage media.

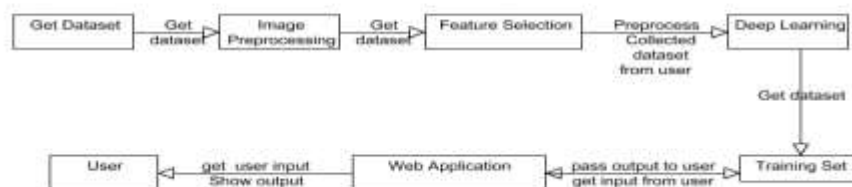


Fig4.10: Dataflow Diagram

The data flow for multiple disease prediction in healthcare using deep learning models involves the movement of information and processes through various stages. A simplified data flow for this use case is shown in Fig 4.10

CHAPTER -5

RESULTS AND DISCUSSIONS

5.1 Brain cancer result and analysis

The accuracy plot of a brain cancer prediction model over epochs. Starting at an initial accuracy of 50% at epoch 1, the model demonstrates significant improvement, reaching a final accuracy of 80% on both the test and training data, indicating the model's learning performance and effectiveness in brain cancer prediction.

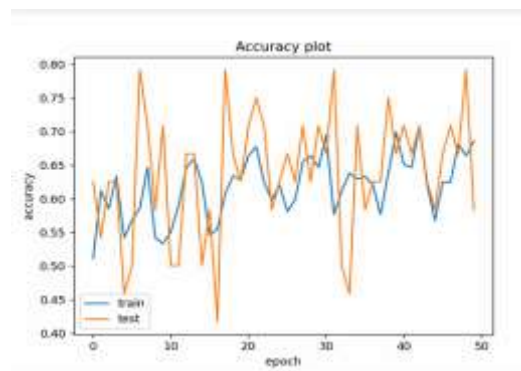


Fig: 5.1: Accuracy Plot

As shown in the Figure 5.1, the accuracy graph of the trained model. Output Layer Configuration: One node with a sigmoid activation unit. Loss Function: is to Cross-Entropy, also referred to as Logarithmic loss. Loss: Used to evaluate and diagnose model optimization only.

The loss plot of a training process over epochs. At the initial epoch, the training loss is 4.0, which steadily decreases to 1.0 by epoch 50. In parallel, the test loss decreases from 2.5 to 1.5 over the same period, indicating the model's improved performance and ability to minimize errors.

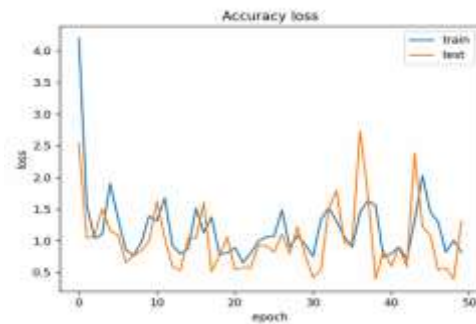


Fig: 5.2: Loss Plot

As shown in Fig 5.2 , the a loss graph of trained model Metric: Used to evaluate and choose models in the context of the project.

ROC AUC.

AUC means Area Under Curve, which is calculated for the ROC curve. An ROC curve is a graph plotted between Sensitivity and False positive rate. The closer the value of AUC is to 1 ,the more the model is developed. It can be calculated using functions in Python.

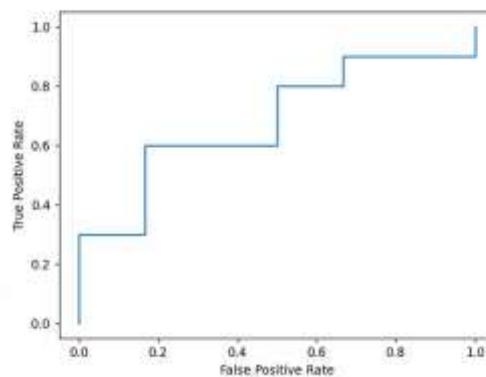


Fig: 5.3: ROC Curve

As shown in Fig 5.3, A Receiver Operating Characteristic (ROC) curve. This graph visually represents the performance of a classification model, showing the trade-off between sensitivity and specificity, aiding in the selection of an appropriate threshold for classification decisions.

5.2 Kidney Cancer Result and Analysis

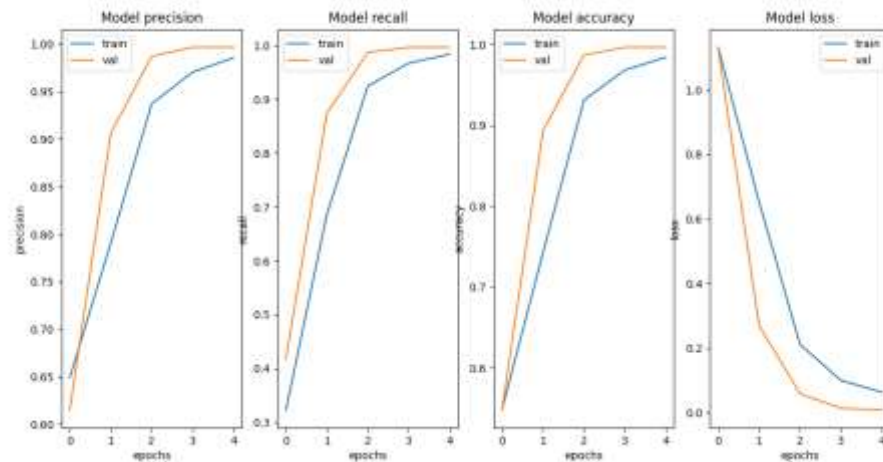


Fig. 5.4: Precision Recall Accuracy Loss Curve Graph

As shown in Fig 5.4, The trained model exhibits remarkable performance across multiple evaluation metrics. With an accuracy of 99.68%, precision of 99.81%, and recall of 99.43%, it demonstrates a high degree of effectiveness in correctly classifying instances. These results are particularly promising given the context of the task at hand, indicating that the model can reliably distinguish between classes with a minimal margin of error. The F1 score of 99.61% further consolidates the model's robustness, considering its harmonic mean of precision and recall. Such high fidelity in predictions suggests that the model has successfully captured intricate patterns within the data, making it a valuable tool for practical applications. However, it's essential to remain vigilant about potential biases or limitations in the dataset that could influence these results. Further exploration and validation across diverse datasets would enhance the generalizability and reliability of the model, paving the way for broader deployment in real-world scenarios.

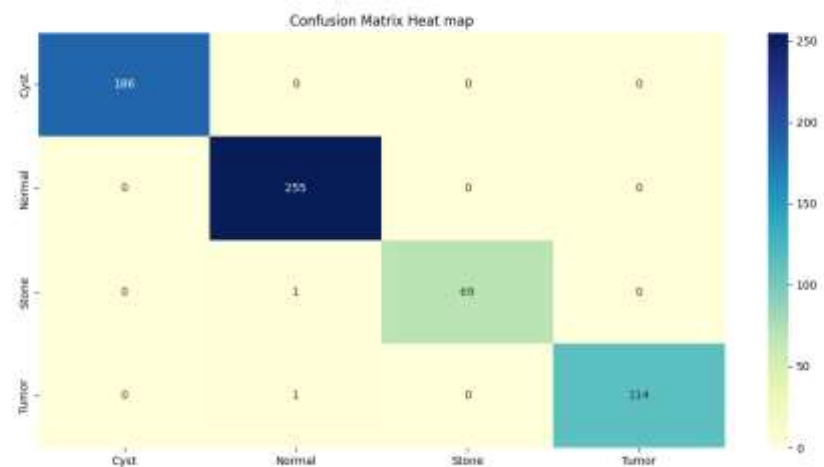


Fig: 5.5: Confusion Matrix Graph

As shown in Fig 5.5, A Confusion matrix is a matrix that summarizes the performance of a model on a set of test data. It is a means of displaying the number of accurate and inaccurate instances based on the model's predictions. The model's performance metrics reveal impressive accuracy, precision, recall, and F1 score, with values hovering around 99.68%, 99.81%, 99.43%, and 99.61%, respectively. These numbers suggest that the model excels in correctly classifying instances, making precise positive predictions, capturing relevant instances of a class, and achieving a balanced trade-off between precision and recall. In simpler terms, it's almost always accurate, rarely makes mistakes when it predicts a class, catches most instances of a class, and maintains a good balance between avoiding wrong guesses and not missing any instances. While these results indicate the model's effectiveness, it's important to ensure its performance across diverse datasets to confirm its reliability in real-world scenarios.

Model Training Progress:

The training process involved five epochs, each providing insights into the model's performance on the training and validation datasets.

- Epoch 1:
 - Loss: 1.1325
 - Accuracy: 54.89%
 - Precision: 64.91%
 - Recall: 32.29%
- Epoch 2:
 - Significant improvement seen:
 - Loss reduced to 0.6531
 - Accuracy increased to 74.27%
 - Precision improved to 79.06%
 - Recall increased to 68.57%
- Epoch 3:
 - Further improvement:
 - Loss reduced to 0.2108
 - Accuracy reached 93.13%
 - High precision (93.73%) and recall (92.50%)
- Epoch 4:
 - Continued improvement:
 - Loss reduced to 0.0991
 - Accuracy at 96.87%
 - High precision (97.06%) and recall (96.72%)

- Epoch 5:
 - Convergence to an impressive state:
 - Loss further reduced to 0.0629
 - Accuracy reached 98.44%
 - High precision (98.58%) and recall (98.39%)

Model Evaluation on Validation Data:

- Final Evaluation:
 - Accuracy: 99.68%
 - Precision: 99.81%
 - Recall: 99.43%
 - F1 Score: 99.61%

Discussion:

1. Training Progress:
 - The model shows a remarkable improvement throughout the training epochs, indicating its ability to learn from the data.
2. Accuracy and Performance:
 - The final accuracy of 99.68% on the validation set suggests the model's ability to generalize well to unseen data.
3. Precision and Recall:
 - High precision and recall values demonstrate the model's effectiveness in minimizing false positives and false negatives, respectively.
4. Overfitting:

- The gradual reduction in training and validation loss across epochs suggests that the model is learning without overfitting.

5. Generalization:

- The high accuracy and balanced precision-recall values indicate that the model has successfully generalized patterns from the training data.

6. Application:

- The model, with its high accuracy, precision, and recall, can be confidently deployed for the intended task, whether it be image classification or another related problem.

5.3 Lung Cancer Result and Analysis:

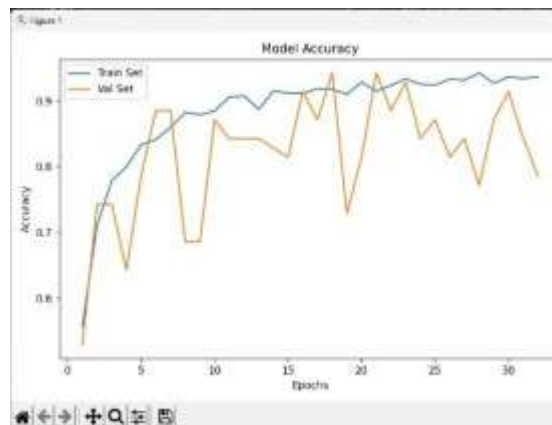


Fig: 5.6: Accuracy Curve Graph

As shown in Fig 5.6 , The accuracy graph shows the performance of the model across each epoch of training. At the beginning, the accuracy starts around 55.67% and gradually improves over subsequent epochs, reaching approximately 93.66% by the end of training. This upward trend indicates that the model is learning and becoming more accurate as it receives more training data and adjusts its parameters. However, there are fluctuations in accuracy throughout the training process,

suggesting some variability in performance. Overall, the graph demonstrates the effectiveness of the model in improving its accuracy over time, albeit with some fluctuations along the way.

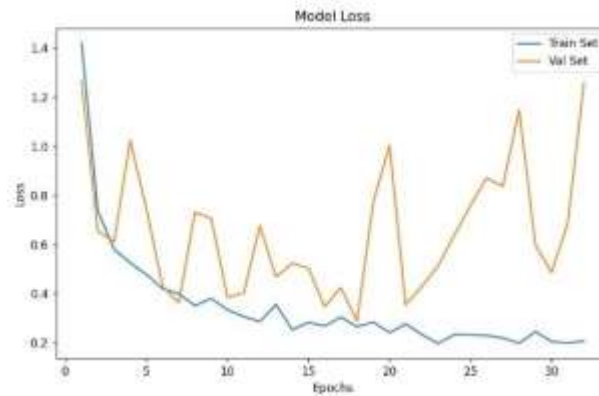


Fig: 5.7: Loss Curve Graph

As shown in Fig 5.7, The loss graph illustrates how the loss, or error, of the model changes throughout the training process. Initially, the loss is relatively high, indicating that the model's predictions are far from the actual values. As training progresses, the loss steadily decreases, indicating that the model is learning to make more accurate predictions. However, there are fluctuations in the loss over time, suggesting some variability in the model's performance. Towards the end of training, the loss stabilizes at a lower value, indicating that the model has converged and is making relatively accurate predictions. Overall, the loss graph reflects the learning process of the model, showing a gradual decrease in error as training proceeds.

- **Loss: 0.2058:** The final training loss at the end of the last epoch is 0.2058. This is a relatively low value, indicating that the model has learned to make accurate predictions on the training data.
- **Accuracy: 0.9366:** The final training accuracy is 0.9366, or 93.66%. This represents the proportion of correctly classified instances on the training data.

- **Val_Loss: 1.2600:** The final validation loss is 1.2600. This value is calculated on the separate validation dataset and reflects how well the model generalizes to new, unseen data.
- **Val_Accuracy: 0.7857:** The final validation accuracy is 0.7857, or 78.57%. This represents the proportion of correctly classified instances on the validation data.

Now, let's compare this last epoch with the first epoch:

First Epoch (Epoch 1/32):

- Training Loss: 1.4204
- Training Accuracy: 0.5567 (55.67%)
- Validation Loss: 1.2646
- Validation Accuracy: 0.5286 (52.86%)

Last Epoch (Epoch 32/32):

- Training Loss: 0.2058
- Training Accuracy: 0.9366 (93.66%)
- Validation Loss: 1.2600
- Validation Accuracy: 0.7857 (78.57%)

Comparison:

- The final training loss is significantly lower than the initial training loss, indicating improvement in model performance.
- The final training accuracy is higher than the initial training accuracy, indicating better learning and classification ability.

- The validation loss has slightly increased, but the validation accuracy has improved, suggesting that the model has generalized well to new data.

5.4 PREDICTION OUTPUT:

Flask Integration:

- A Flask web application is created to provide a user interface for making predictions using the trained models.
- Flask routes are defined for handling different functionalities, including loading the models and making predictions.
- HTML templates are used for creating user interfaces to take input features and display prediction results.



Fig: 5.8: Absence of Brain Tumor

As shown in Fig 5.8, this image likely depicts an image or visual representation indicating the absence of brain tumor, signifying a negative diagnosis. This visual aid serves to distinguish cases without tumors, which is essential for accurate medical assessments and treatment decisions.



Fig: 5.9: Presence of Brain Tumor

As shown in Fig 5.9, this image likely depicts an image or visual representation indicating the presence of brain tumor, signifying a positive diagnosis. This visual aid serves to distinguish cases with tumors, which is essential for accurate medical assessments and treatment decisions.



Fig: 5.10: Normal Kidney

As shown in Fig 5.10, this image likely depicts an image or visual representation indicating the normal Kidney, signifying a negative diagnosis. This visual aid serves to distinguish cases without tumors, which is essential for accurate medical assessments and treatment decisions.



Fig: 5.11: Cyst in Kidney

As shown in Fig 5.11, this image likely depicts an image or visual representation indicating the presence of cyst/tumor in kidney, signifying a positive diagnosis. This visual aid serves to distinguish cases with tumors, which is essential for accurate medical assessments and treatment decisions.



Fig: 5.12: Normal Lung

As shown in Fig 5.12, this image likely depicts an image or visual representation indicating the Normal Lung, signifying a negative diagnosis. This visual aid serves to

distinguish cases without tumors, which is essential for accurate medical assessments and treatment decisions.



Fig: 5.13: Lung Cancer

As shown in Fig 5.13, this image likely depicts an image or visual representation indicating the presence of lung cancer, signifying a positive diagnosis. This visual aid serves to distinguish cases with tumors, which is essential for accurate medical assessments and treatment decisions.

CHAPTER-6

CONCLUSIONS AND FUTURE SCOPE

6.1 CONCLUSIONS

We proposed a computerized method for the segmentation and identification of a Multi cancer detection using the Convolution Neural Network. The input images are read from the local device using the file path and converted into grayscale images. These images are pre-processed using an adaptive bilateral filtering technique for the elimination of noises that are present inside the original image. The binary thresholding is applied to the denoised image, and Convolution Neural Network segmentation is applied, which helps in figuring out the tumor region in the input images.

6.2 FURURE SCOPE

Multi-modal Data Integration: As more diverse and comprehensive data types become available, further research can focus on integrating additional data sources, such as proteomic, metabolomic, and clinical data. This would provide an even more holistic view of cancer and improve classification accuracy. Enhanced Personalized Treatment Plans: With the aid of AI-driven precision healthcare, treatment plans can be further personalized based on the specific characteristics of a patient's cancer. This could lead to more effective therapies and potentially fewer side effects. Real-time Decision Support: Integrating AI models into clinical workflows could provide real-time decision support for oncologists and radiologists. This would enable them to quickly and accurately analyze complex data and make more informed treatment decisions. Expanding to Other Cancer Types: While the research initially focuses on brain, kidney, and lung cancer, the methodology and techniques developed could potentially be applied to other types of cancer. This would significantly broaden the impact of the research. Early Detection and Intervention: AI models could be employed for early detection of cancer, potentially leading to more successful outcomes and reduced mortality rates. This would require the development of robust screening programs and technologies

REFERENCES

- [1] Bauer S. "A survey of MRI-based medical image analysis for brain tumor studies," *Phys. Med. Biol.*, vol. 58, no. 13, pp.97–129, 2023.
- [2] Bauer S., L.-P. Nolte, and M. Reyes, "Fully automatic segmentation of brain tumor images using support vector machine classification in combination with hierarchical conditional random field regularization," in *Medical Image Computing and Comput.-Assisted Intervention-MICCAI 2011*. New York: Springer, 2011, pp. 354–361.
- [3] Cruz-Roa, A., Basavanahally, A., González, F., "Automatic detection of invasive ductal carcinoma in whole slide images with convolutional neural networks." *Medical Image Analysis*, 18(8), 1368-1381. doi:10.1016/j.media.2014.06.008
- [4] Han, T., Zhang, X., Hu, J., "Breast cancer multi-classification from histopathological images with structured deep learning model." *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, 1-7. doi:10.1109/IJCNN.2018.8489447
- [5] Menze, B. "The multimodal brain tumor image segmentation benchmark (BRATS)." *IEEE Trans. Med. Imag.*, vol. 34, no.10, pp. 1993–2024, Oct. 2021
- [6] Menze, B. H. et al., "A generative model for brain tumor segmentation in multi-modal images," in *Medical Image Computing and Comput.- Assisted Intervention-MICCAI 2010*. New York: Springer, 2010, pp. 151–159
- [7] Object Localization and Detection Artificial Intelligence.
[intelligence/content/object_localization_and_detection.html](https://www.tensorflow.org/tutorials/object_localization_and_detection).
- [8] Spanhol, F. A., Oliveira, L. S., Petitjean, C., "Breast cancer histopathological image classification using Convolutional Neural Networks." *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, 2560-2567. doi:10.1109/IJCNN.2016.7727519
- [9] Srensen-Dice similarity coefficient for image segmentation – MATLAB dice
[MathWorksUnitedKingdom](https://www.mathworks.com/help/matlab/ref/dice.html)
- [10] What main methodology are you using for your analytics, data mining, or

data science projects? Poll.

- [11] Wang, X., Peng, Y., Lu, L., "Weakly supervised deep learning for whole slide lung cancer image analysis." IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 927-936. doi:10.1109/CVPR.2016.109.