

# **Application of Deep Learning in Medical Image Classification and Segmentation**

## **A Dissertation**

**Submitted by**

**A Shriya(19GANSE001)**

**Harishankara H V(19GANSE017)**

**Kushal B (19GANSE024)**

**Mohammed Danish Rabbani (19GANSE028)**

**8<sup>th</sup> Semester B. Tech (ISE)**



**Department of Computer Science and Engineering**

**Under the Guidance of**

**Dr. Thriveni J**

**Professor, Dept of CSE, UVCE**

**Bangalore University**

**UNIVERSITY VISVESVARAYA COLLEGE OF ENGINEERING**

**K R Circle Bengaluru – 560001**

**June 2023**

## **ACKNOWLEDGEMENTS**

The satisfaction of completing any task would be incomplete without mentioning the people who made it possible, whose constant guidance and encouragement helped us reach success. First and foremost, we would like to express our sincere gratitude and respect to the organization University of Visvesvaraya College of Engineering, Bangalore, for providing us an opportunity to carry out our project work.

We thank **Dr. Jayakara S M**, Vice Chancellor, Bangalore University and **Dr. H N Ramesh**, Principal, University of Visvesvaraya College of Engineering, Bangalore, for providing us with all the facilities that helped us to carry out the work easily. We wish to place our gratitude thanks to **Dr. Vimala H S**, Professor and Chairperson, Department of Computer Science and Engineering, UVCE, who helped us make our project a great success. We would like to express our deepest thanks to **Dr. Thriveni J**, Professor of the Department of Computer Science and Engineering, for her constant guidance and support as an internal guide with continuous guidance and encouragement to complete the project successfully on time.

Lastly, we extend our thanks to all the teaching and non-teaching fraternity in the Department of Computer Science and Engineering, for always being helpful over the years. We are very grateful to our parents, friends and well-wisher for their continuous moral support and encouragement.

**A Shriya**

**19GANSE001**

**Harishankara H V**

**19GANSE017**

**Kushal S B**

**19GANSE024**

**Mohammed Danish Rabbani**

**19GANSE028**

## **ABSTRACT**

Big medical data mainly include electronic health record data, medical image data, gene information data, etc. Among them, medical image data account for the vast majority of medical data at this stage. How to apply big medical data to clinical practice? This is an issue of great concern to medical and computer researchers, and intelligent imaging and deep learning provides a good answer. This project introduces the application of intelligent imaging and deep learning in the field of big data analysis and early diagnosis of diseases, combining the latest research progress of big data analysis of medical images and the work of our team in the field of big data analysis of medical images, especially the classification and segmentation of medical images. Deep learning has developed into a hot research field, and there are dozens of algorithms, each with its own advantages and disadvantages. These algorithms cover almost all aspects of our image processing, which mainly focus on classification and segmentation.

Ensembling is a technique in machine learning where multiple models are combined to make predictions. By leveraging the strengths of different models and reducing individual model errors, ensembling can lead to improved accuracy. The diversity of models helps capture different aspects of the data, and the ensemble can average out biases or mistakes made by individual models, resulting in more robust and accurate predictions. Ensembling also helps reduce the risk of overfitting, enhances stability, and leverages the "wisdom of the crowd" principle. Overall, ensembling is a powerful technique that combines the predictions of multiple models to achieve higher accuracy than using a single model alone. We have achieved better accuracy for classification and segmentation of medical images using various ensemble methods.

# CONTENTS

I.	Acknowledgements.....	i
II.	Abstract.....	ii
III.	Contents.....	iii
IV.	List of Figures.....	v
V.	List of Tables.....	x
1.	INTRODUCTION .....	1
	1.1 Overview .....	1
	1.2 Medical Image Segmentation.....	2
	1.2.1 Brain Tumor Segmentation.....	2
	1.3 Medical Image Classification.....	3
	1.3.1 Acute Lymphoblastic Leukemia.....	3
	1.3.2 Brain Cancer.....	4
	1.3.3 Breast Cancer.....	4
	1.3.4 Cervical Cancer.....	4
	1.3.5 Kidney Cancer.....	5
	1.3.6 Lung Cancer.....	5
	1.3.7 Colon Cancer.....	5
	1.3.8 Lymphoma.....	5
	1.3.9 Oral Cancer.....	6
2.	LITERATURE SURVEY.....	7
3.	SYSTEM DESIGN.....	26
	3.1 System Design for Tumor Segmentation.....	26
	3.2 System Design for Classification.....	27
4.	SYSTEM ARCHITECTURE.....	28
	4.1 System Architecture for Brain Tumor Segmentation.....	28
	4.2 System Architecture for Classification.....	30
5.	SYSTEM REQUIREMENTS.....	34
	5.1 Hardware Requirements.....	34
	5.2 Software Requirements.....	34
6.	IMPLEMENTATION.....	38
	6.1 Implementation of Segmentation Models.....	38

6.1.1	Dataset.....	38
6.1.2	Data Transforms.....	40
6.1.3	Modeling.....	41
6.1.3.1	SEGRESNET Architecture.....	41
6.1.3.2	UNETR Architecture.....	43
6.1.4	Training.....	45
6.1.4.1	Ensemble.....	46
6.2	Implementation of Classification Models.....	46
6.2.1	Classification of Different Types of Cancer.....	44
6.2.2	DenseNet201 Architecture.....	50
6.2.3	MobileNetV3Small Architecture.....	52
6.2.4	ResNet50v2 Architecture.....	54
6.2.5	VGG19 Architecture.....	57
6.2.6	Training.....	59
6.2.7	Ensemble.....	62
7.	PERFORMANCE METRICS.....	64
7.1.	Metrics Used for Segmentation.....	64
7.2.	Metrics Used for Classification.....	65
8.	PERFORMANCE ANALYSIS.....	70
8.1	Result for Brain Tumor Segmentation.....	70
8.2	Result for Classification Models.....	73
9.	CONCLUSION.....	86
10.	REFERENCES.....	87

## LIST OF FIGURES

Fig 3.1	System design for brain tumor segmentation	26
Fig 3.2	System design for classification	27
Fig 4.1	Glioma sub-regions	28
Fig 4.2	Multi cancer dataset from Kaggle	30
Fig 6.1	The common workflow of predefined datasets	39
Fig 6.2	Brain MRI images	39
Fig 6.3	BRATS 3D segmentation	39
Fig 6.4	Segmentation of brain matter	40
Fig 6.5	Schematic visualization of the network architecture	42
Fig 6.6	Architecture of each block	42
Fig 6.7	Overview of the UNETR architecture	45
Fig 6.8	Loss function	45
Fig 6.9	Optimizer function	45
Fig 6.10	Ensemble	46
Fg 6.11	Acute Lymphoblastic Leukemia	46

Fig 6.12	Brain Cancer	47
Fig 6.13	Breast Cancer	47
Fig 6.14	Cervical Cancer	48
Fig 6.15	Kidney Cancer	48
Fig 6.16	Lung Cancer	49
Fig 6.17	Lymphoma	49
Fig 6.18	Oral Cancer	50
Fig 6.19	DenseNet Architecture	52
Fig 6.20	MobileNet Architecture	54
Fig 6.21	Example of Residual Block	56
Fig 6.22	Residual Blocks	56
Fig 6.23	RESNETv2	57
Fig 6.24	VGG19 Architecture	59
Fig 6.25	Ensemble Technique Overview	63
Fig 7.1	Confusion Matrix	66
Fig 7.2	Accuracy	67

Fig 7.3	Precision	68
Fig 7.4	Recall	68
Fig 7.5	F1 Score	69
Fig 8.1	Original Image	70
Fig 8.2	SEGRESNET - Ground Truth	70
Fig 8.3	SEGRESNET - Output	70
Fig 8.4	Original Image	71
Fig 8.5	UNETR - Ground Truth	71
Fig 8.6	UNETR - Output	71
Fig 8.7	Ground Truth	72
Fig 8.8	Model Output	72
Fig 8.9	Result for Acute Lymphoblastic Leukemia using DenseNet201	74
Fig 8.10	Result for Acute Lymphoblastic Leukemia using MobileNetv3Small	74
Fig 8.11	Result for Acute Lymphoblastic Leukemia using VGG19	74
Fig 8.12	Result for Acute Lymphoblastic Leukemia using ResNet50V2	75
Fig 8.13	Result for Acute Lymphoblastic Leukemia using Ensemble	75
Fig 8.14	Result for Brain Cancer using DenseNet201	75
Fig 8.15	Result for Brain Cancer using MobileNetv3Small	76
Fig 8.16	Result for Brain Cancer using VGG19	76
Fig 8.17	Result for Brain Cancer using ResNet50V2	76

Fig 8.18	Result for Brain Cancer using Ensemble	76
Fig 8.19	Result for Breast Cancer using DenseNet201	77
Fig 8.20	Result for Breast Cancer using MobileNetv3Small	77
Fig 8.21	Result for Breast Cancer using VGG19	77
Fig 8.22	Result for Breast Cancer using ResNet50V2	77
Fig 8.23	Result for Breast Cancer using Ensemble	78
Fig 8.24	Result for Cervical Cancer using DenseNet201	78
Fig 8.25	Result for Cervical Cancer using MobileNetv3Small	79
Fig 8.26	Result for Cervical Cancer using VGG19	79
Fig 8.27	Result for Cervical Cancer using ResNet50V2	79
Fig 8.28	Result for Cervical Cancer using Ensemble	79
Fig 8.29	Result for Kidney Cancer using DenseNet201	80
Fig 8.30	Result for Kidney Cancer using MobileNetv3Small	80
Fig 8.31	Result for Kidney Cancer using VGG19	80
Fig 8.32	Result for Kidney Cancer using ResNet50V2	80
Fig 8.33	Result for Kidney Cancer using Ensemble	81
Fig 8.34	Result for Lung & Colon Cancer using DenseNet201	81
Fig 8.35	Result for Lung & Colon Cancer using MobileNetv3Small	81
Fig 8.36	Result for Lung & Colon Cancer using VGG19	82
Fig 8.37	Result for Lung & Colon Cancer using ResNet50V2	82

Fig 8.38	Result for Lung & Colon Cancer using Ensemble	82
Fig 8.39	Result for Lymphoma using DenseNet201	83
Fig 8.40	Result for Lymphoma using MobileNetv3Small	83
Fig 8.41	Result for Lymphoma using VGG19	83
Fig 8.42	Result for Lymphoma using ResNet50V2	83
Fig 8.43	Result for Lymphoma using Ensemble	84
Fig 8.44	Result for Oral Cancer using DenseNet201	84
Fig 8.45	Result for Oral Cancer using MobileNetv3Small	85
Fig 8.46	Result for Oral Cancer using VGG19	85
Fig 8.47	Result for Oral Cancer using ResNet50V2	85
Fig 8.48	Result for Oral Cancer using Ensemble	85

## LIST OF TABLES

Table 8.1	Result for Acute Lymphoblastic Leukemia using DenseNet201	74
Table 8.2	Result for Brain Cancer using DenseNet201	75
Table 8.3	Result for Breast Cancer using DenseNet201	76
Table 8.4	Result for Cervical Cancer using DenseNet201	78
Table 8.5	Result for Kidney Cancer using DenseNet201	80
Table 8.6	Result for Lung & Colon Cancer using DenseNet201	81
Table 8.7	Result for Lymphoma using DenseNet201	82
Table 8.8	Result for Oral Cancer using DenseNet201	84

# **CHAPTER 1**

## **INTRODUCTION**

### **1.1 Overview**

Since 2006, deep learning has emerged as a branch of the machine learning field in people's field of vision. It is a method of data processing using multiple layers of complex structures or multiple processing layers composed of multiple nonlinear transformations. In recent years, deep learning has made breakthroughs in the fields of computer vision, speech recognition, natural language processing, audio recognition and bioinformatics. Deep learning has been praised as one of the top ten technological breakthroughs since 2013 due to its considerable application prospects in data analysis. The deep learning method simulates the human neural network. By combining multiple nonlinear processing layers, the original data is abstracted layer by layer, and different levels of abstract features are obtained from the data and used for target detection, classification or segmentation. The advantage of deep learning is to replace the manual acquisition feature with unsupervised or semi-supervised feature learning and hierarchical feature extraction efficient algorithms.

Medical care is about the health of people. At present, the amount of medical data is huge, but it is crucial to make good use of this huge medical data to contribute to the medical industry. Although the amount of medical data is huge, there are still many problems: medical data is diverse, including maps, texts, videos, magnets, etc.; due to different equipment used, the quality of data varies greatly; data presents fluctuating characteristics, over time and specific events change; due to differences in individuals, the law of the disease has no universal applicability. There are many factors that cannot be dealt with in the existence of these problems. Medical imaging is a very important part of medical data.

Deep learning can be used for various purposes with respect to medical datasets, such as medical image segmentation and classification.

---

In this project we have developed models for brain tumor segmentation and classification of tumors of multiple cancers such as Acute Lymphoblastic Leukemia, Brain Cancer, Breast Cancer, Cervical Cancer, Kidney Cancer, Lung and Colon Cancer, Lymphoma and Oral Cancer.

## **1.2 Medical Image Segmentation**

Medical image segmentation allows for the precise delineation and identification of structures and abnormalities within medical images. It helps physicians and radiologists accurately locate and analyze specific regions of interest, such as tumors, organs, blood vessels, or pathologies. This accurate diagnosis is crucial for effective treatment planning and monitoring of diseases. Deep learning algorithms can automate the image segmentation process, reducing the time and effort required by medical professionals to manually analyze and segment medical images. This automation enhances the efficiency of the diagnostic workflow and enables radiologists to focus more on interpreting results and making informed decisions. Accurate segmentation of medical images allows for personalized treatment planning. By precisely delineating the target region, such as a tumor, radiation oncologists can design treatment plans that deliver optimal radiation doses to the tumor while sparing healthy surrounding tissues. This helps minimize the side effects of treatment and improves patient outcomes.

### **1.2.1 Brain Tumor Segmentation**

Brain tumor localization and segmentation from magnetic resonance imaging (MRI) are hard and important tasks for several applications in the field of medical analysis. Brain tumors include the most threatening types of tumors around the world. Glioma, the most common primary brain tumors, occurs due to the carcinogenesis of glial cells in the spinal cord and brain. Glioma is characterized by several histological and malignancy grades, and an average survival time of fewer than 14 months after diagnosis for glioblastoma patients. However, the manual segmentation and analysis of structural MRI images of brain tumors is an arduous and time-consuming task which, thus far, can only be accomplished by professional neuroradiologists.

Therefore, an automatic and robust brain tumor segmentation will have a significant impact on brain tumor diagnosis and treatment. Furthermore, it can also lead to timely diagnosis and treatment of neurological disorders such as Alzheimer's disease (AD), schizophrenia, and dementia. Here we have segmented the tumor and classified the parts of the tumor(The whole tumor,Tumor core,Enhancing tumor).

## **1.3 Medical Image Classification**

Deep learning-based image classification algorithms can analyze medical images and accurately classify them into different categories, such as healthy or diseased, benign or malignant, or specific disease subtypes. This automation can assist medical professionals in the diagnosis process, providing a preliminary assessment and guiding further investigations or treatments. Deep learning algorithms can help detect subtle patterns or abnormalities in medical images that may not be easily discernible to the human eye. By accurately classifying images, these algorithms can aid in the early detection of diseases or conditions, enabling timely intervention and improving patient outcomes. For example, in the case of cancer, early detection can significantly increase the chances of successful treatment. Medical image classification using deep learning can serve as a decision support tool for healthcare providers. By providing insights and predictions based on image analysis, these algorithms can assist in treatment planning, risk assessment, and patient management. This can help physicians make informed decisions, improve clinical workflows, and enhance patient care.

### **1.3.1 Acute Lymphoblastic Leukemia**

Acute Lymphoblastic Leukemia (ALL), also known as acute lymphocytic leukemia, is a type of cancer that affects the blood and bone marrow. It is characterized by the rapid production of immature white blood cells, specifically lymphoblasts, which are abnormal cells that do not mature into healthy white blood cells. These lymphoblasts accumulate in the bone marrow and interfere with the production of normal blood cells. The definitive diagnosis of acute lymphoblastic leukemia (ALL), as a highly prevalent cancer, requires invasive, expensive, and time-consuming diagnostic tests. ALL diagnosis using peripheral blood smear (PBS) images plays a vital role in the

initial screening of cancer from non-cancer cases. The examination of these PBS images by laboratory users is riddled with problems such as diagnostic error because the nonspecific nature of ALL signs and symptoms often leads to misdiagnosis.

The subtypes of ALL are Benign, Early, Pre, Pro. Deep learning models can be used to classify the Blood smear images of ALL patients into the subtypes of ALL.

### **1.3.2 Brain Cancer**

Brain cancer, also known as brain tumor or intracranial tumor, refers to the abnormal and uncontrolled growth of cells in the brain. It is a serious condition that can affect individuals of any age.

Common types of tumors are:

- Gliomas: Gliomas are the most common type of brain tumors, originating from the glial cells that support and nourish neurons.
- Meningiomas: Meningiomas develop from the meninges, the protective membranes surrounding the brain and spinal cord.
- Pituitary adenomas: These tumors form in the pituitary gland, a small gland located at the base of the brain. They can disrupt hormone production and have varying degrees of malignancy.

Deep learning models are used to classify the type of tumor from the medical image.

### **1.3.3 Breast Cancer**

Breast cancer is a type of cancer that develops in the cells of the breast. It is one of the most common types of cancer among women worldwide, but it can also affect men.

Breast tumors can be classified as Benign or malignant using deep learning models.

### **1.3.4 Cervical Cancer**

Cervical cancer is a type of cancer that develops in the cells of the cervix, which is the lower part of the uterus that connects to the vagina. It is primarily caused by persistent infection with high-risk types of the human papillomavirus (HPV), a sexually transmitted infection. Cervical cancer is one of the most preventable and treatable forms of cancer when detected early through screening and vaccination. Cervical

---

cancer screening often involves analyzing Pap smear or colposcopy images. Machine learning algorithms can be trained to classify these images as normal or abnormal, helping identify potential precancerous or cancerous lesions. Techniques like convolutional neural networks (CNNs) have been successfully used for image classification tasks.

### **1.3.5 Kidney Cancer**

Kidney cancer, also known as renal cancer, refers to the development of cancerous tumors in the kidneys. It is one of the most common types of cancer, and it primarily affects adults.

Deep learning models can be used to predict if the tumor is present or not.

### **1.3.6 Lung Cancer**

Lung cancer refers to the uncontrolled growth of abnormal cells in the tissues of the lungs. It is the leading cause of cancer-related deaths worldwide for both men and women.

Deep learning models can be used to classify among the following types-Lung Adenocarcinoma, Lung Benign Tissue, Lung Squamous Cell Carcinoma.

### **1.3.7 Colon Cancer**

Colon cancer, also known as colorectal cancer, refers to the development of cancerous cells in the colon or rectum. It is one of the most common types of cancer worldwide.

Several factors may increase the risk of developing colon cancer, including age (most cases occur after age 50), a personal or family history of colorectal cancer or polyps, inflammatory bowel disease (such as Crohn's disease or ulcerative colitis), certain inherited gene mutations (such as Lynch syndrome or familial adenomatous polyposis), a sedentary lifestyle, obesity, smoking, and a diet high in red and processed meats.

Deep learning model can be used to classify among the following types-Colon Adenocarcinoma and Colon Benign Tissue

### **1.3.8 Lymphoma**

---

Lymphoma is a type of cancer that affects the lymphatic system, which is a part of the body's immune system. It involves the abnormal growth of cells in the lymphocytes, a type of white blood cell.

Deep learning models can be used to classify among the following types-Chronic Lymphocytic Leukemia, Follicular Lymphoma and Mantle Cell Lymphoma.

### **1.3.9 Oral Cancer**

Oral cancer, also known as mouth cancer or oral cavity cancer, refers to the development of malignant cells in the oral cavity, which includes the lips, tongue, gums, inner lining of the cheeks, floor of the mouth, and the hard and soft palate. It is a type of head and neck cancer.

Deep learning models can be used to classify among the following types - Normal and Oral Squamous Cell Carcinoma.

## CHAPTER 2

### LITERATURE SURVEY

- [1] Abadi, Martin et al. (2016). ‘Tensorflow: Large-scale machine learning on heterogeneous distributed systems’. In: arXiv preprint arXiv:1603.04467

TensorFlow is a widely-used open-source software library for machine learning developed by the Google Brain team. It was first released in 2015 and has since become one of the most popular machine learning libraries. The paper "TensorFlow: Large-scale machine learning on heterogeneous distributed systems" describes the design and implementation of TensorFlow and its capabilities for large-scale machine learning on heterogeneous distributed systems. The paper also discusses the use of dataflow graphs, which are a central concept in TensorFlow, for representing computations and the use of a high-performance execution engine for executing those computations. The paper concludes by discussing the scalability and performance of TensorFlow and its applications in various domains.

- [2] Abraham, Nabila and Naimul Mefraz Khan (2019). ‘A novel focal tversky loss function with improved attention u-net for lesion segmentation’. In: 2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019). IEEE

The paper "A novel focal Tversky loss function with improved attention U-net for lesion segmentation" presents a new approach for lesion segmentation using a deep learning model. The model is based on the U-net architecture, which is a type of convolutional neural network (CNN) used for image segmentation tasks. The authors propose to improve the U-net model by incorporating a new loss function called the focal Tversky loss function and an attention mechanism. The focal Tversky loss function is an extension of the popular Tversky loss function, which is used to measure the similarity between the predicted and ground truth segmentation masks. The attention mechanism is used to focus the network's attention on the regions of the image that are most important for the task of lesion segmentation. The authors

---

evaluate the proposed model on two publicly available datasets and show that it outperforms the standard U-net model and other state-of-the-art methods.

- [3] Anithadevi, D and K Perumal (2016). 'A hybrid approach based segmentation technique for brain tumor in MRI Images'. In: arXiv preprint arXiv:1603.02447

The paper "A hybrid approach based segmentation technique for brain tumor in MRI images" presents a new method for the segmentation of brain tumors in magnetic resonance imaging (MRI) images. The authors propose a hybrid approach that combines the advantages of two different methods: a region-based method and a boundary-based method. The region-based method uses a level set algorithm to segment the tumor region, while the boundary-based method uses a snake algorithm to segment the tumor boundary. The authors evaluate the proposed method on a dataset of MRI images and compare it with other state-of-the-art methods. They found that the proposed method outperforms other methods in terms of accuracy and computational efficiency. The paper concludes by discussing the potential applications of the proposed method in the diagnosis and treatment of brain tumors.

- [4] Hatamizadeh, A., Tang, Y., Nath, V., Yang, D., Myronenko, A., Landman, B., Roth, H.R. and Xu, D., 2022. Unetr: Transformers for 3d medical image segmentation. In Proceedings of the IEEE/CVF winter conference on applications of computer vision (pp. 574-584).

The paper titled "UNETR: Transformers for 3D Medical Image Segmentation" proposes a novel approach to 3D medical image segmentation using transformers. Medical image segmentation is an essential task in various clinical applications, such as tumor detection and organ delineation. Traditional methods for image segmentation rely on convolutional neural networks (CNNs), but transformers have shown great success in natural language processing tasks.

---

The authors introduce UNETR, a framework that combines the popular U-Net architecture with transformers for 3D medical image segmentation. U-Net is a widely used CNN-based architecture that has achieved remarkable results in medical image segmentation. However, it struggles with capturing long-range dependencies in the image, which transformers excel at. Transformers have a self-attention mechanism that allows them to attend to any part of the input, making them suitable for capturing global context information.

In the proposed UNETR framework, the authors replace the traditional U-Net encoder with a transformer encoder. They leverage the advantages of transformers to capture long-range dependencies and extract rich contextual information from the 3D medical images. The transformer encoder consists of multiple layers, each containing a multi-head self-attention mechanism and position-wise feed-forward networks. The output of the transformer encoder is then passed to the decoder, which follows the U-Net architecture. The decoder progressively upsamples the features and combines them with the corresponding features from the encoder to produce the final segmentation map.

To train UNETR, the authors employ a combination of supervised and self-supervised learning. They use a large-scale dataset of 3D medical images and manually annotated ground truth segmentation maps for supervised training. Additionally, they propose a self-supervised pretraining step using a contrastive loss, where the model learns to distinguish between positive and negative pairs of augmented images.

Experimental results demonstrate that UNETR outperforms traditional CNN-based approaches for 3D medical image segmentation. It achieves state-of-the-art performance on several benchmark datasets, including the BraTS dataset for brain tumor segmentation. The authors also conduct ablation studies to investigate the impact of different components in the UNETR framework, showing the effectiveness of the transformer encoder in capturing long-range dependencies.

---

In conclusion, the paper presents UNETR, a novel framework that combines transformers with the U-Net architecture for 3D medical image segmentation. By leveraging the self-attention mechanism of transformers, UNETR achieves superior performance compared to traditional CNN-based methods. The proposed approach has the potential to improve various clinical applications by enabling more accurate and efficient segmentation of 3D medical images.

- [5] Myronenko, A., 2019. 3D MRI brain tumor segmentation using autoencoder regularization. In Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries: 4th International Workshop, BrainLes 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 16, 2018, Revised Selected Papers, Part II 4 (pp. 311-320). Springer International Publishing.

The paper titled "3D MRI Brain Tumor Segmentation Using Autoencoder Regularization" introduces a method for segmenting brain tumors in 3D magnetic resonance imaging (MRI) scans. Accurate tumor segmentation is crucial for diagnosis, treatment planning, and monitoring of brain tumor patients. The proposed approach combines a convolutional neural network (CNN) with an autoencoder regularization technique to improve segmentation performance.

The authors start by training an initial CNN model for tumor segmentation using a labeled dataset. However, they observe that the CNN tends to produce inaccurate segmentations and struggles with capturing fine details due to limited training data and the complex nature of tumors. To address this issue, they introduce an autoencoder regularization framework.

The autoencoder regularization framework consists of an encoder network and a decoder network. The encoder network takes the input MRI scan and encodes it into a lower-dimensional latent space representation. The decoder network then reconstructs the original input from the latent space representation. The idea behind the autoencoder regularization is that by forcing the CNN's feature representations to be compressible and

---

reconstructible by the autoencoder, it encourages the network to learn more meaningful and discriminative features.

During training, the authors employ a two-step process. In the first step, they train the CNN using the labeled dataset as usual. In the second step, they freeze the CNN weights and jointly train the encoder and decoder of the autoencoder regularization framework. The loss function consists of two components: a reconstruction loss that measures the dissimilarity between the input and the reconstructed output, and a regularization loss that encourages the CNN's feature representations to be compressible.

To evaluate the proposed method, the authors conduct experiments on a publicly available brain tumor segmentation challenge dataset called BraTS. They compare their approach against other state-of-the-art methods. The results demonstrate that the combination of the CNN and autoencoder regularization significantly improves tumor segmentation performance. The proposed method achieves higher Dice scores, which measure the overlap between the predicted and ground truth tumor segmentations, indicating more accurate segmentations.

In conclusion, the paper presents a 3D MRI brain tumor segmentation method that combines a CNN with an autoencoder regularization framework. By leveraging the regularization effect of the autoencoder, the proposed approach enhances the CNN's ability to capture fine details and improve segmentation accuracy. The experimental results on the BraTS dataset demonstrate the effectiveness of the method, highlighting its potential for aiding clinical decision-making and improving patient care in brain tumor diagnosis and treatment.

- 
- [6] Gu, Shanqing, Manisha Pednekar and Robert Slater (2019). 'Improve Image Classification Using Data Augmentation and Neural Networks'.

The paper "Improve Image Classification Using Data Augmentation and Neural Networks" presents a method to improve image classification accuracy using data augmentation and neural networks. Data augmentation is a technique where new training samples are generated by applying various transformations to the original images, such as rotation, scaling, and flipping. This can help to prevent overfitting and improve the robustness of the model. The authors propose to use data augmentation techniques to generate more training samples from the original images, and use a neural network to classify the images.

The authors evaluate their method on several benchmark datasets and compare it to other state-of-the-art methods. They show that their method, which combines data augmentation and neural networks, can significantly improve the classification accuracy compared to other methods. The paper also highlights the importance of preprocessing the data, specifically by removing the noise present in the images and normalizing the data to improve the results. Additionally, the authors also show the impact of increasing the number of augmented images generated with the help of data augmentation and the improvement in results with an increase in the number of augmented images.

- 
- [7] Salçin, Kerem et al. (2019). 'Detection and classification of brain tumours from MRI images using faster R-CNN'.

The paper "Detection and classification of brain tumours from MRI images using faster R-CNN" presents a method for detecting and classifying brain tumors from magnetic resonance imaging (MRI) images using the Faster R-CNN algorithm. Faster R-CNN is a popular object detection algorithm that combines a region proposal network (RPN) and a convolutional neural network (CNN) to detect and classify objects in images. The authors propose to use Faster R-CNN to detect and classify brain tumors in MRI images.

---

The authors evaluate their method on a dataset of MRI images and compare it to other state-of-the-art methods. They show that the proposed method outperforms other methods in terms of both detection and classification accuracy. Additionally, the authors also proposed a new dataset that they used to evaluate the proposed method. The dataset contains 535 MRI images and their corresponding ground truth labels. The dataset was made publicly available for the research community, which will help other researchers to use the dataset to evaluate their own methods and compare their results with the proposed method.

- [8] Jonas Prellberg, Oliver Kramer (2019), ‘Acute Lymphoblastic Leukemia Detection from Microscopic Images Using Weighted Ensemble of Convolutional Neural Networks’

Acute Lymphoblastic Leukemia (ALL) is a blood cell cancer characterized by numerous immature lymphocytes. Even though automation in ALL prognosis is an essential aspect of cancer diagnosis, it is challenging due to the morphological correlation between malignant and normal cells. This article has automated the ALL detection task from microscopic cell images, employing deep Convolutional Neural Networks (CNNs). The authors explore the weighted ensemble of different deep CNNs to recommend a better ALL cell classifier. The weights for the ensemble candidate models are estimated from their corresponding metrics, such as accuracy, F1-score, AUC, and kappa values. Various data augmentations and pre-processing are incorporated for achieving a better generalization of the network. The authors utilize the publicly available C-NMC-2019 ALL dataset to conduct all the comprehensive experiments. Their proposed weighted ensemble model, using the kappa values of the ensemble candidates as their weights, has outputted a weighted F1-score of 88.6 %, a balanced accuracy of 86.2 %, and an AUC of 0.941 in the preliminary test set. The qualitative results displaying the gradient class activation maps confirm that the introduced model has a concentrated learned region. In contrast, the ensemble candidate models, such as Xception,

1 arXiv:2105.03995v1 [eess.IV] 9 May 2021 VGG-16, DenseNet-121, MobileNet, and InceptionResNet-V2, separately produce coarse and scatter learned areas for most example cases. Since the proposed kappa value-based weighted ensemble yields a better result for the aimed task in this article, it can be experimented in other domains of medical diagnostic applications.

- [9] Marco Aurélio Granero, Cristhian Xavier Hernández, Marcos Eduardo Valle (2021) ‘Quaternion-Valued Convolutional Neural Network Applied for Acute Lymphoblastic Leukemia Diagnosis’

The field of neural networks has seen significant advances in recent years with the development of deep and convolutional neural networks. Although many of the current works address real-valued models, recent studies reveal that neural networks with hypercomplex-valued parameters can better capture, generalize, and represent the complexity of multidimensional data. This paper explores the quaternion-valued convolutional neural network application for a pattern recognition task from medicine, namely, the diagnosis of acute lymphoblastic leukemia. Precisely, we compare the performance of real-valued and quaternion-valued convolutional neural networks to classify lymphoblasts from the peripheral blood smear microscopic images. The quaternion-valued convolutional neural network achieved better or similar performance than its corresponding real-valued network but using only 34% of its parameters. This result confirms that quaternion algebra allows capturing and extracting information from a color image with fewer parameters.

- [10] Md Zahangir Alom, Chris Yakopcic, Tarek M. Taha, Vijayan K. Asari · ‘Breast Cancer Classification from Histopathological Images with Inception Recurrent Residual Convolutional Neural Network’

The Deep Convolutional Neural Network (DCNN) is one of the most powerful and successful deep learning approaches. DCNNs have already provided

superior performance in different modalities of medical imaging including breast cancer classification, segmentation, and detection. Breast cancer is one of the most common and dangerous cancers impacting women worldwide. In this paper, we have proposed a method for breast cancer classification with the Inception Recurrent Residual Convolutional Neural Network (IRRCNN) model. The IRRCNN is a powerful DCNN model that combines the strength of the Inception Network (Inception-v4), the Residual Network (ResNet), and the Recurrent Convolutional Neural Network (RCNN). The IRRCNN shows superior performance against equivalent Inception Networks, Residual Networks, and RCNNs for object recognition tasks. In this paper, the IRRCNN approach is applied for breast cancer classification on two publicly available datasets including BreakHis and Breast Cancer Classification Challenge 2015. The experimental results are compared against the existing machine learning and deep learning-based approaches with respect to image-based, patch-based, image-level, and patient-level classification. The IRRCNN model provides superior classification performance in terms of sensitivity, Area Under the Curve (AUC), the ROC curve, and global accuracy compared to existing approaches for both datasets.

[11] Aditya Golatkar, Deepak Anand, Amit Sethi ‘Classification of Breast Cancer Histology using Deep Learning’

Breast Cancer is a major cause of death worldwide among women. Hematoxylin and Eosin (H&E) stained breast tissue samples from biopsies are observed under microscopes for the primary diagnosis of breast cancer. In this paper, we propose a deep learning-based method for classification of H&E stained breast tissue images released for BACH challenge 2018 by fine-tuning Inception-v3 convolutional neural network (CNN) proposed by Szegedy et al. These images are to be classified into four classes namely, i) normal tissue, ii) benign tumor, iii) in-situ carcinoma and iv) invasive carcinoma. Our strategy is to extract patches based on nuclei density instead of random or grid sampling, along with rejection of patches that are not rich in nuclei

(non-epithelial) regions for training and testing. Every patch (nuclei-dense region) in an image is classified in one of the four above mentioned categories. The class of the entire image is determined using majority voting over the nuclear classes. The authors obtained an average four class accuracy of 85% and an average two class (non-cancer vs. carcinoma) accuracy of 93%, which improves upon a previous benchmark by Araujo et al.

[12] Rishav Pramanik, Momojit Biswas, et al - ‘A fuzzy distance-based ensemble of deep models for cervical cancer detection’.

**Background and Objective** Cervical cancer is one of the leading causes of women’s death. Like any other disease, cervical cancer’s early detection and treatment with the best possible medical advice are the paramount steps that should be taken to ensure the minimization of after-effects of contracting this disease. PaP smear images are one the most effective ways to detect the presence of such type of cancer. This article proposes a fuzzy distance-based ensemble approach composed of deep learning models for cervical cancer detection in PaP smear images. The authors employ three transfer learning models for this task: Inception V3, MobileNet V2, and Inception ResNet V2, with additional layers to learn data-specific features. To aggregate the outcomes of these models, they propose a novel ensemble method based on the minimization of error values between the observed and the ground-truth. For samples with multiple predictions, they first take three distance measures, i.e., Euclidean, Manhattan (City-Block), and Cosine, for each class from their corresponding best possible solution. They then defuzzify these distance measures using the product rule to calculate the final predictions. Results In the current experiments, we have achieved 95.30%, 93.92%, and 96.44% respectively when Inception V3, MobileNet V2, and Inception ResNet V2 run individually. After applying the proposed ensemble technique, the performance reaches 96.96% which is higher than the individual models. Conclusion Experimental outcomes on three publicly available datasets ensure that the proposed model presents competitive results compared to

state-of-the-art methods. The proposed approach provides an end-to-end classification technique to detect cervical cancer from PaP smear images. This may help the medical professionals for better treatment of cervical cancer. Thus increasing the overall efficiency in the whole testing process.

[13] “Oral Cancer Detection using Machine Learning and Deep Learning Techniques”  
Nanditha B R, Geetha Kiran A Sanathkumar M P

The paper titled "Oral Cancer Detection using Machine Learning and Deep Learning Techniques" discusses the application of machine learning and deep learning techniques for the detection of oral cancer. Oral cancer is a significant health concern, and early detection plays a crucial role in improving patient outcomes.

The authors explore various techniques to develop an automated system for oral cancer detection. They start by acquiring a dataset of oral images, which includes both normal and cancerous samples. The dataset is preprocessed to enhance image quality and remove noise.

The authors then extract relevant features from the images using traditional machine learning algorithms. These features include color, texture, and shape descriptors. They employ techniques such as Principal Component Analysis (PCA) and Local Binary Patterns (LBP) to reduce the dimensionality of the feature space. The reduced feature set is used to train classifiers, such as Support Vector Machines (SVM) and Random Forests, to distinguish between normal and cancerous oral images.

In addition to traditional machine learning techniques, the authors investigate the effectiveness of deep learning methods for oral cancer detection. They employ Convolutional Neural Networks (CNNs), which have shown remarkable performance in various image classification tasks. The CNN is trained on the oral image dataset, leveraging its ability to automatically learn

---

relevant features from the images. The trained CNN is then used to classify new oral images as normal or cancerous.

The authors evaluate the performance of the machine learning and deep learning models using various metrics, including accuracy, sensitivity, specificity, and area under the Receiver Operating Characteristic (ROC) curve. They compare the results of different techniques to assess their effectiveness in oral cancer detection.

The experimental results demonstrate that both machine learning and deep learning techniques can effectively detect oral cancer. The deep learning approach, particularly the CNN-based model, achieves superior performance compared to traditional machine learning algorithms. The CNN is capable of capturing intricate patterns and features from the oral images, leading to more accurate and reliable cancer detection.

In conclusion, the paper highlights the potential of machine learning and deep learning techniques for oral cancer detection. The combination of these techniques with image analysis provides a promising approach to automate the process of oral cancer diagnosis. Early detection through automated systems can significantly improve patient outcomes by enabling timely interventions and treatments.

---

[14] Panwar, S.A., Munot, M.V., Gawande, S. and Deshpande, P.S., 2021. A reliable and an efficient approach for diagnosis of brain tumor using transfer learning. *Biomed Pharmacol J*, 14(1), pp.283-294.

The paper titled "A Reliable and an Efficient Approach for Diagnosis of Brain Tumor using Transfer Learning" presents a method for the diagnosis of brain tumors using transfer learning. Brain tumor diagnosis is a critical task in medical imaging, and the proposed approach aims to improve the reliability and efficiency of the diagnosis process. The authors start by acquiring a large dataset of brain MRI scans, which includes both tumor and non-tumor cases.

They preprocess the images to enhance their quality and ensure consistency. To overcome the limitations of limited data and computational resources, the authors employ transfer learning. Transfer learning allows the model to leverage pre-trained neural network architectures and knowledge from large-scale image datasets. In this case, the authors utilize pre-trained models, such as VGG16 and ResNet, which are trained on natural image datasets like ImageNet. The authors adapt the pre-trained models for brain tumor diagnosis by fine-tuning them on the acquired brain MRI dataset. Fine-tuning involves training the pre-trained models on the specific task of tumor classification using the brain MRI images. The weights of the pre-trained models are updated during this process to learn the discriminative features relevant to brain tumor detection. To further enhance the reliability of the diagnosis, the authors propose an ensemble learning strategy. They train multiple fine-tuned models using different pre-trained architectures and combine their predictions to make the final diagnosis. The ensemble approach helps to mitigate the effects of model bias and variance, resulting in more accurate and robust predictions. The authors evaluate the performance of their approach using various evaluation metrics, including accuracy, sensitivity, specificity, and F1-score. They compare the results of their method with traditional machine learning algorithms and other deep learning approaches to showcase its superiority. The experimental results demonstrate that the proposed transfer learning-based approach achieves reliable and efficient brain tumor diagnosis. The ensemble of fine-tuned models outperforms individual models and traditional machine learning algorithms. The approach achieves high accuracy and other favorable metrics, indicating its potential for real-world clinical applications.

In conclusion, the paper presents a reliable and efficient approach for brain tumor diagnosis using transfer learning. By leveraging pre-trained models and fine-tuning them on a specific brain MRI dataset, the proposed method effectively learns discriminative features for tumor detection. The ensemble learning strategy further enhances the reliability of the diagnosis. The

experimental results validate the effectiveness of the approach, highlighting its potential to assist clinicians in accurate and efficient brain tumor diagnosis.

[15] Ullah, N., Khan, J.A., Khan, M.S., Khan, W., Hassan, I., Obayya, M., Negm, N. and Salama, A.S., 2022. An Effective Approach to Detect and Identify Brain Tumors Using Transfer Learning. *Applied Sciences*, 12(11), p.5645.

The paper titled "An Effective Approach to Detect and Identify Brain Tumors Using Transfer Learning" introduces a method for the detection and identification of brain tumors using transfer learning. The authors aim to develop an efficient and accurate approach to assist in the diagnosis of brain tumors. The authors begin by assembling a large dataset of brain MRI scans, consisting of both tumor and non-tumor cases. These images undergo preprocessing to enhance their quality and normalize their characteristics. To overcome the limitations of limited data and computational resources, the authors employ transfer learning. Transfer learning allows the utilization of pre-trained deep neural network architectures and knowledge from large-scale image datasets. In this study, pre-trained models such as VGG16 and InceptionV3, which were trained on general image datasets, are employed. The authors fine-tune the pre-trained models on the brain MRI dataset. Fine-tuning involves training the pre-trained models on the specific task of tumor detection and identification using the brain MRI images. By doing so, the models learn to recognize the distinctive features associated with brain tumors. In addition to fine-tuning, the authors propose a region-based approach to further refine the tumor detection. They segment the brain MRI images into smaller regions of interest and classify each region individually. This approach enables more precise localization and identification of tumor regions within the brain scans. The authors evaluate the performance of their approach using various metrics, including accuracy, sensitivity, specificity, and F1-score. They compare their results with other traditional machine learning methods and deep learning approaches to highlight the effectiveness of their proposed method. The experimental results demonstrate that the transfer learning-based

approach achieves high accuracy and reliable tumor detection and identification. The fine-tuned models outperform traditional machine learning algorithms and other deep learning techniques. The region-based approach enhances the localization of tumor regions, leading to more accurate identification.

In conclusion, the paper presents an effective approach for brain tumor detection and identification using transfer learning. By leveraging pre-trained models and fine-tuning them on a brain MRI dataset, the proposed method effectively learns to recognize brain tumors. The region-based approach further improves the localization and identification accuracy. The experimental results validate the effectiveness of the approach, indicating its potential for assisting medical professionals in diagnosing and treating brain tumors.

[16] Dalia Alzu’bi,Malak Abdullah, et.al, “Kidney Tumor Detection and Classification Based on Deep Learning Approaches: A New Dataset in CT Scans” [2022]

Kidney tumor (KT) is one of the diseases that have affected our society and is the seventh most common tumor in both men and women worldwide. Compared to the tedious and time-consuming traditional diagnosis, automatic detection algorithms of deep learning (DL) can save diagnosis time, improve test accuracy, reduce costs, and reduce the radiologist’s workload. In this paper, we present detection models for diagnosing the presence of KTs in computed tomography (CT) scans. Toward detecting and classifying KT, we proposed 2D-CNN models; three models are concerning KT detection such as a 2D convolutional neural network with six layers (CNN-6), a ResNet50 with 50 layers, and a VGG16 with 16 layers. The last model is for KT classification as a 2D convolutional neural network with four layers (CNN-4). In addition, a novel dataset from the King Abdullah University Hospital (KAUH) has been collected that consists of 8,400 images of 120 adult patients who have performed CT scans for suspected kidney masses. The dataset was divided into 80% for the training set and 20% for the testing set. The accuracy results

for the detection models of 2D CNN-6 and ResNet50 reached 97%, 96%, and 60%, respectively. At the same time, the accuracy results for the classification model of the 2D CNN-4 reached 92%. Their novel models achieved promising results; they enhance the diagnosis of patient conditions with high accuracy.

- [17] Md. Alamin Talukder, Md. Manowarul Islam, et.al, “Machine Learning-based Lung and Colon Cancer Detection using Deep Feature Extraction and Ensemble Learning”[2022]

Cancer is a fatal disease caused by a combination of genetic diseases and a variety of biochemical abnormalities. Lung and colon cancer have emerged as two of the leading causes of death and disability in humans. The histopathological detection of such malignancies is usually the most important component in determining the best course of action. Early detection of the ailment on either front considerably decreases the likelihood of mortality. Machine learning and deep learning techniques can be utilized to speed up such cancer detection, allowing researchers to study a large number of patients in a much shorter amount of time and at a lower cost. In this research work, the authors introduced a hybrid ensemble feature extraction model to efficiently identify lung and colon cancer. It integrates deep feature extraction and ensemble learning with high-performance filtering for cancer image datasets. The model is evaluated on histopathological (LC25000) lung and colon datasets. According to the study findings, their hybrid model can detect lung, colon, and (lung and colon) cancer with accuracy rates of 99.05%, 100%, and 99.30%, respectively. The study's findings show that their proposed strategy outperforms existing models significantly. Thus, these models could be applicable in clinics to support the doctor in the diagnosis of cancers.

- [18] Satvik Garg, Somya Garg, “Prediction of lung and colon cancer through analysis of histopathological images by utilizing Pre-trained CNN models with visualization of class activation and saliency maps”[2021]

This paper intends to utilize and alter the current pre-trained CNN-based model to identify lung and colon cancer utilizing histopathological images with better augmentation techniques. In this paper, eight distinctive Pre-trained CNN models, VGG16, NASNetMobile, InceptionV3, InceptionResNetV2, ResNet50, Xception, MobileNet, and DenseNet169 are trained on LC25000 dataset. The model performances are assessed on precision, recall, f1score, accuracy, and auroc score. The results exhibit that all eight models accomplished noteworthy results ranging from 96% to 100% accuracy. Subsequently, GradCAM and SmoothGrad are also used to picture the attention images of Pre-trained CNN models classifying malignant and benign images.

[19] Maria Frucci, Daniel Riccio, et.al, “A Deep Learning Approach for Breast Invasive Ductal Carcinoma Detection and Lymphoma Multi-Classification in Histological Images”

This paper explores deep learning methods for the automatic analysis of Hematoxylin and Eosin stained histological images of breast cancer and lymphoma. In particular, a deep learning approach is proposed for two different use cases: the detection of invasive ductal carcinoma in breast histological images and the classification of lymphoma sub-types. Both use cases have been addressed by adopting a residual convolutional neural network that is part of a convolutional autoencoder network (i.e., FusionNet). The performances have been evaluated on the public datasets of digital histological images and have been compared with those obtained by using different deep neural networks (UNet and ResNet). Additionally, comparisons with the state of the art have been considered, in accordance with different deep learning approaches. The experimental results show an improvement of 5.06% in F-measure score for the detection task and an improvement of 1.09% in the accuracy measure for the classification task.

---

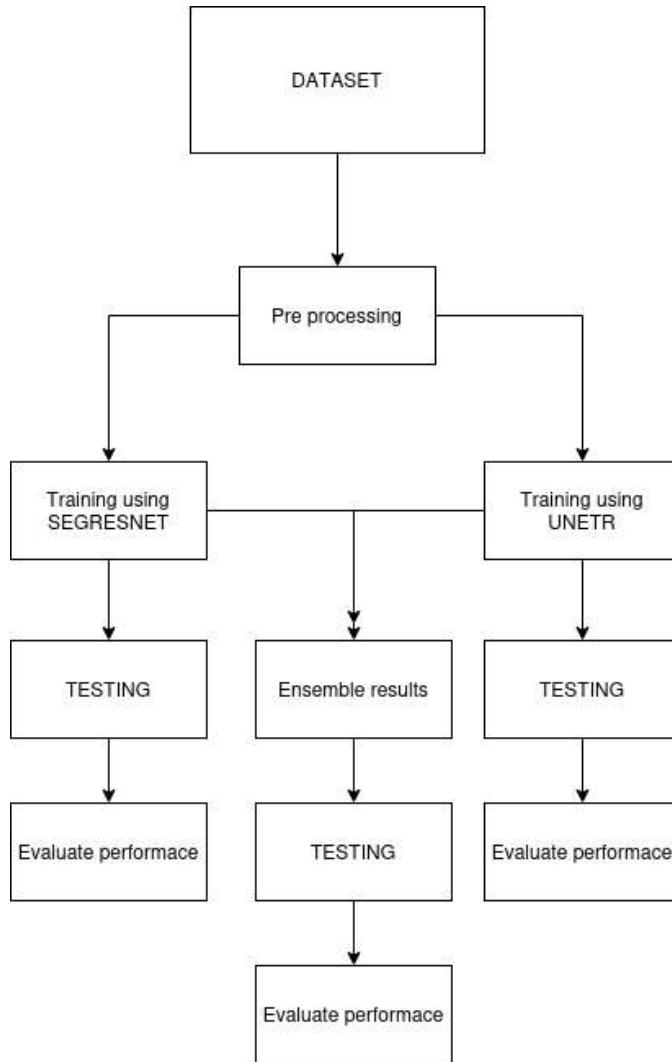
[20] Zeyad Ghaleb Al-Mekhlafi, Ebrahim Mohammed Senan, et.al, “Diagnosis of Histopathological Images to Distinguish Types of Malignant Lymphomas Using Hybrid Techniques Based on Fusion Features”

Because of the similarity of the morphological characteristics between lymphoma types, this study aimed to extract features using various algorithms and deep learning models and combine them together into feature vectors. Two datasets have been applied, each with two different systems for the reliable diagnosis of malignant lymphoma. The first system was a hybrid system between DenseNet-121 and ResNet-50 to extract deep features and reduce their dimensions by the principal component analysis (PCA) method, using the support vector machine (SVM) algorithm for classifying low-dimensional deep features. The second system was based on extracting the features using DenseNet-121 and ResNet-50 and combining them with the hand-crafted features extracted by gray level co-occurrence matrix (GLCM), fuzzy color histogram (FCH), discrete wavelet transform (DWT), and local binary pattern (LBP) algorithms and classifying them using a feed-forward neural network (FFNN) classifier. All systems achieved superior results in diagnosing the two datasets of malignant lymphomas. An FFNN classifier with features of ResNet-50 and hand-crafted features reached an accuracy of 99.5%, specificity of 100%, sensitivity of 99.33%, and AUC of 99.86% for the first dataset. In contrast, the same technique reached 100% for all measures to diagnose the second dataset.

## CHAPTER 3

### SYSTEM DESIGN

#### 3.1 System Design for Brain Tumor Segmentation



**Fig 3.1 System design for brain tumor segmentation**

Dataset contains various 3D MRI Images. Dataset is loaded and transformed (random flips and rotation). Pre-processed data is used to train 2 models (SEGRESNET and UNETR). Both the models are tested using dice score as the performance metrics.

In the ensemble step, we try to ensemble results of both the models on testing data based on mean pixel value.

### 3.2 System Design for Classification

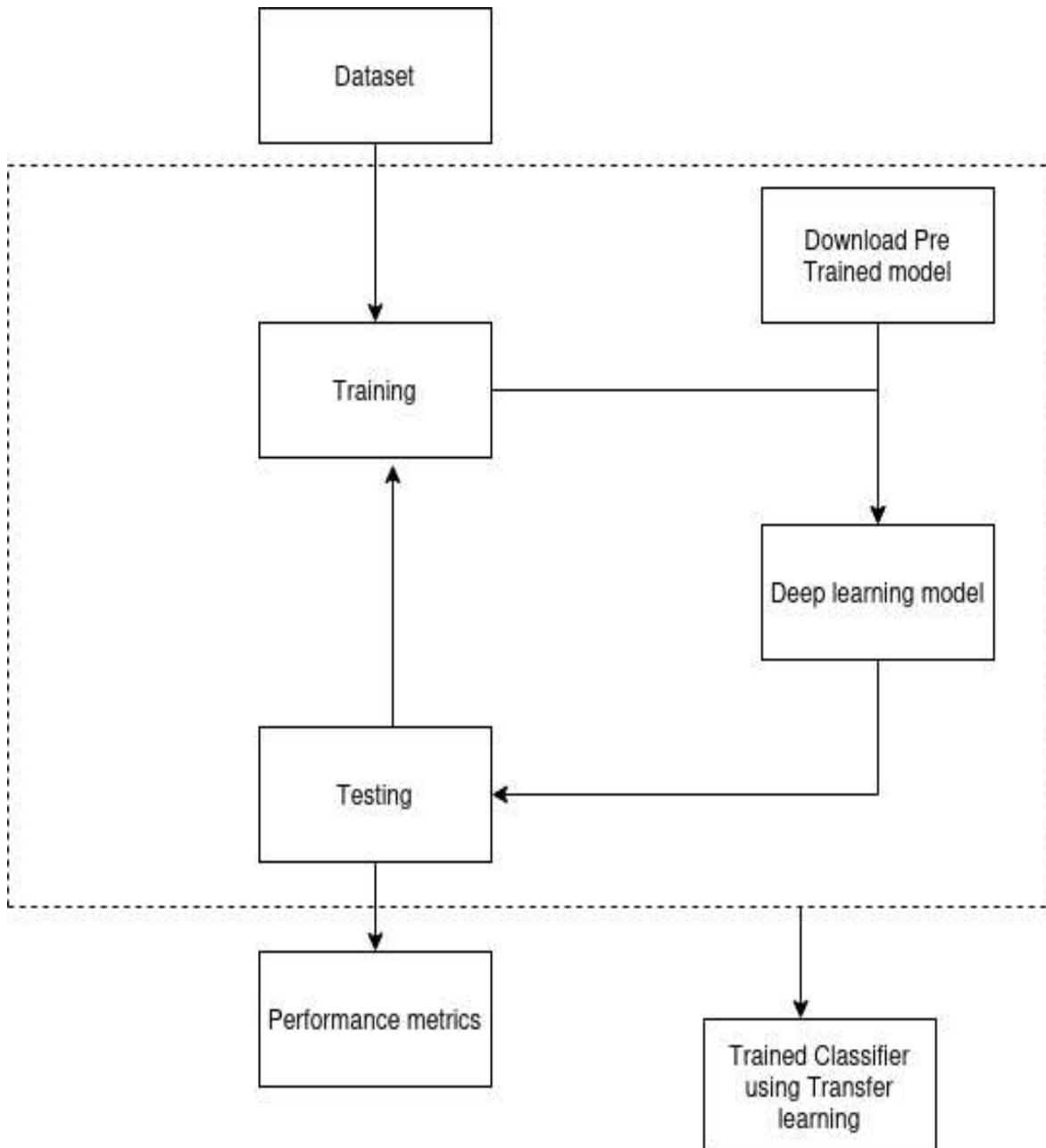


Fig 3.2 System design for classification

---

Transfer learning is used to classify the medical images for various cancers.

Dataset for each cancer is loaded and pre-processed to a particular image size and intensity. Various pre-trained models are downloaded and trained with this data. After the training, these models are tested with a testing dataset and performance metrics (accuracy of the model) is calculated and evaluated.

## CHAPTER 4

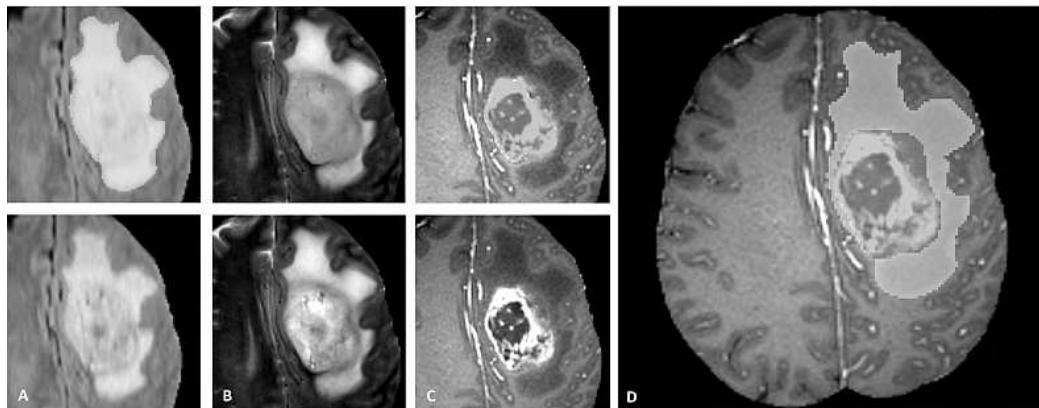
### SYSTEM ARCHITECTURE

#### 4.1 System Architecture for Brain Tumor Segmentation

We propose to build a deep learning method to segment tumors and its part from an MRI image of the brain.

##### Dataset

The dataset comes from medical segmentation decathlon. The dataset contains 750 4D volumes (484 Training + 266 Testing) MRI images. The dataset consists of the following types-FLAIR, T1w, T1gd,T2w. Dataset sources are BRATS 2016 and 2017 datasets.



**Fig 4.1 Glioma sub-regions**

The image patches show from left to right:

1. the whole tumor (yellow) visible in T2-FLAIR (Fig.A).
2. the tumor core (red) visible in T2 (Fig.B).
3. the enhancing tumor structures (light blue) visible in T1Gd, surrounding the cystic/necrotic components of the core (green) (Fig. C).
4. The segmentations are combined to generate the final labels of the tumor sub-regions (Fig.D): edema (yellow), non-enhancing solid core (red), necrotic/cystic core (green), enhancing core (blue).

### **Pre-Processing**

Each MRI has 3 label images (Ground truth). The labels are:-

- Label 1 is the peritumoral edema
- Label 2 is the GD-enhancing tumor
- Label 3 is the necrotic and non-enhancing tumor core

These labels are converted to brats classes which are TC (Tumor core), WT (Whole tumor) and ET (Enhancing tumor).

Label 2 and label 3 are merged to construct the tumor core.

Label 1, label 2 and label 3 are merged to construct Whole tumor.

Label 2 is enhancing tumors.

Each Image is cropped to a specific dimension.

### **Data Augmentation**

Transform the data by randomly adjusting the intensity and producing rotations and flips of the MRI image.

### **Modeling**

We propose a deep learning method to segment the tumor in the following classes-TC (Tumor core), WT (Whole tumor) and ET (Enhancing tumor).

We propose to design 2 architectures (2 models) to segment the tumor and then combine the results from both the models to get a better tumor prediction.

Convolutional neural networks are very similar to the ordinary feed-forward neural networks. They differ in the sense that CNNs assume explicitly that the inputs are images, which enables us to encode specific properties in the architecture to recognize certain patterns in the images. The CNNs make use of spatial nature of the data. It means, CNNs perceive objects similar to our perception of different objects in nature. For example, we recognize various objects by their shapes, size and colors. These

objects are combinations of edges, corners, color patches, etc. CNNs can use a variety of detectors (such as edge detectors, corner detectors) to interpret images. These detectors are called filters or kernels. The mathematical operator that takes an image and a filter as input and produces a filtered output (e.g. edges, corners, etc.) is called convolution.

First architecture is based on SegResNet which is an encoder-decoder based semantic segmentation network based on ResNet blocks with Instance normalization and deep supervision in the decoder branch. It is used for brain tumor segmentation.

Second architecture is based on UNETR, or UNet Transformer, which is a Transformer-based architecture for medical image segmentation that utilizes a pure transformer as the encoder to learn sequence representations of the input volume – effectively capturing the global multi-scale information.

## **4.2 System Architecture for Classification**

Cancer	Classes	Images
Acute Lymphoblastic Leukemia	4	20000
Brain Cancer	3	15000
Breast Cancer	2	10000
Cervical Cancer	5	25000
Kidney Cancer	2	10000
Lung and Colon Cancer	5	25000
Lymphoma	3	15000
Oral Cancer	2	10000

**Fig 4.2 Multi cancer dataset from Kaggle**

### **Dataset**

Dataset been used is available on Kaggle (Multi cancer Dataset). This Dataset contains Images of various types of Cancer.

Details of the dataset are:-

- Each Subclass contains 5000 Images
- Each image is of size 512X512

### **Preprocessing**

Preprocessing is an essential step in preparing data for machine learning tasks, including deep learning models. It involves transforming and manipulating the raw input data to improve model performance, reduce noise, handle missing values, and ensure compatibility with the chosen model architecture.

- Data normalization: Normalizing the input data is crucial to ensure that features have similar scales and distributions. This step helps prevent certain features from dominating others during training. Common normalization techniques are rescaling features to a specific range (e.g., [0, 1]) or standardizing them to have zero mean and unit variance.
- Image preprocessing: For image data, preprocessing steps may include resizing images to a consistent resolution, cropping or padding to a specific aspect ratio, and converting images to a standardized color space (e.g., RGB or grayscale). Additionally, you may need to subtract mean pixel values or apply other image-specific transformations to improve the model's performance.

### **Modeling**

The use of traditional machine learning algorithms along with hand engineered feature extraction is the most popular method for modeling among papers. We propose to use 4 deep learning models and then use the model that is the most accurate for a particular type of cancer to classify the tumor.

CNN, or Convolutional Neural Network, is a type of deep learning model specifically designed for analyzing grid-like data, such as images and videos. CNNs are widely used in computer vision tasks, including image classification, object detection, image segmentation, and more.

For each cancer type above we have trained 4 Models using transfer learning techniques.

The key characteristic of CNNs is their ability to automatically learn hierarchical representations of data through convolutional and pooling layers.

---

Following are the models:-

The first model is ResNet. A Residual Neural Network (a.k.a. Residual Network, ResNet) is a deep learning model in which the weight layers learn residual functions with reference to the layer inputs. A Residual Network is a network with skip connections that perform identity mappings, merged with the layer outputs by addition. It behaves like a Highway Network whose gates are opened through strongly positive bias weights. This enables deep learning models with tens or hundreds of layers to train easily and approach better accuracy when going deeper.

The second model is MobileNet. MobileNet is a computer vision model open-sourced by Google and designed for training classifiers. It uses depthwise convolutions to significantly reduce the number of parameters compared to other networks, resulting in a lightweight deep neural network. MobileNet is Tensorflow's first mobile computer vision model.

The third model is VGGNet. VGG stands for Visual Geometry Group; it is a standard deep Convolutional Neural Network (CNN) architecture with multiple layers. The “deep” refers to the number of layers with VGG-16 or VGG-19 consisting of 16 and 19 convolutional layers. The VGG architecture is the basis of ground-breaking object recognition models. Developed as a deep neural network, the VGGNet also surpasses baselines on many tasks and datasets beyond ImageNet. Moreover, it is now still one of the most popular image recognition architectures.

The fourth model is DenseNet. DenseNet is a network architecture where each layer is directly connected to every other layer in a feed-forward fashion (within each *dense block*). For each layer, the feature maps of all preceding layers are treated as separate inputs whereas its own feature maps are passed on as inputs to all subsequent layers. This connectivity pattern yields state-of-the-art accuracies on CIFAR10/100 (with or without data augmentation) and SVHN. On the large scale ImageNet dataset, DenseNet achieves a similar accuracy as ResNet, but using less than half the amount of parameters and roughly half the number of FLOPs.

---

We proposed to ensemble all the predictions of these models according to the confidence level of the model for each type of Cancer in order to achieve better predictions on the subclass for each class.

## CHAPTER 5

# SYSTEM REQUIREMENTS

### 5.1 Hardware Requirements

- Intel i7 or higher
- At least 16 GB ram
- 12 GB GPU

### 5.2 Software Requirements

#### **Python 3.0 or higher:**

Python is an interpreted high-level general-purpose programming language. Its design philosophy emphasizes code readability with its use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

#### **Jupyter:**

Project Jupyter is a project and community whose goal is to "develop open-source software, open-standards, and services for interactive computing across dozens of programming languages"

#### **VSCode:**

Visual Studio Code is a source-code editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git.

---

**Python Libraries that were used are:**

**Pandas:**

pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series.

**Torchvision:**

Torchvision is a library for Computer Vision that goes hand in hand with PyTorch. It has utilities for efficient Image and Video transformations, some commonly used pre-trained models, and some datasets.

**Matplotlib:**

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt

**Numpy:**

NumPy, which stands for Numerical Python, is a library consisting of multidimensional array objects and a collection of routines for processing those arrays. Using NumPy, mathematical and logical operations on arrays can be performed.

**Sklearn:**

Scikit-learn is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and

---

statistical modeling including classification, regression, clustering and dimensionality reduction via a consistent interface in Python.

**Keras:**

Keras is a deep learning API written in Python, running on top of the machine learning platform TensorFlow. It was developed with a focus on enabling fast experimentation. Being able to go from idea to result as fast as possible is key to doing good research. Keras is the high-level API of the TensorFlow platform: an approachable, highly-productive interface for solving machine learning problems, with a focus on modern deep learning. It provides essential abstractions and building blocks for developing and shipping machine learning solutions with high iteration velocity.

**Tensorflow:**

It is a software library for numerical computation using data flow graphs where nodes in the graph represent mathematical operations and edges in the graph represent the multidimensional data arrays (called tensors) communicated between them.

**PyTorch:**

PyTorch is an open source machine learning library based on the Torch library, used for applications such as computer vision and natural language processing, primarily developed by Facebook's AI Research lab. It is free and open-source software.

**Monai:**

MONAI is a set of open-source, freely available collaborative frameworks built for accelerating research and clinical collaboration in Medical Imaging. It is built on top of PyTorch and is released under the Apache 2.0 license. MONAI provides domain-specific capabilities for training AI models for

---

healthcare imaging. MONAI Core is the flagship library of Project MONAI and provides domain-specific capabilities for training AI models for healthcare imaging. MONAI Deploy aims to become the de-facto standard for developing packaging, testing, deploying, and running medical AI applications in clinical production.

## CHAPTER 6

# IMPLEMENTATION

## 6.1 Implementation of Segmentation Models

We have trained 2 models which segment the brain tumor and classify the tumor regions. 2 models are based on SEGRESNET and UNETR architecture respectively.

After the models are trained we have written to ensemble the results of the two models to achieve better segmentation of each sub-class.

### 6.1.1 Dataset

Challenge data set. The MSD data set is publicly available under a Creative Commons license CC-BY-SA4.0, allowing broad (including commercial) use. The training data used in this study is available at <http://medicaldecathlon.com/>.

Modality: Multimodal multisite MRI data (FLAIR, T1w, T1gd,T2w)

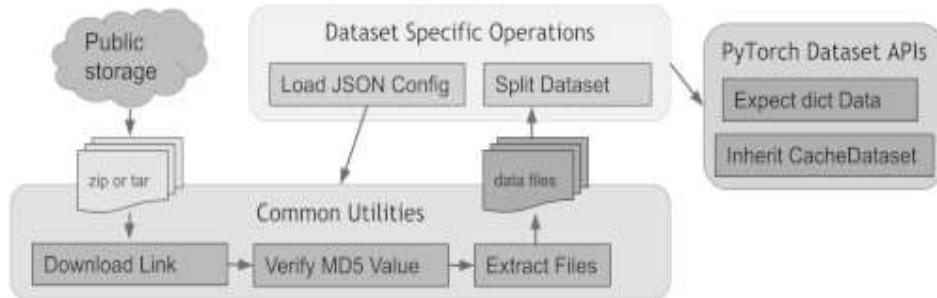
Dataset for task 1 in the challenge Dataset is used for our purpose.

The data set consists of 484 multiparametric-magnetic resonance images (mp-MRI) from patients diagnosed with either glioblastoma or lower-grade glioma. The sequences used were native T1-weighted (T1), post-Gadolinium (Gd) contrast T1-weighted (T1-Gd), native T2-weighted (T2), and T2 Fluid-Attenuated Inversion Recovery (FLAIR). The corresponding target ROIs were the three tumor sub-regions, namely edema, enhancing, and non-enhancing tumor. This data set was selected due to the challenge of locating these complex and heterogeneously-located targets. The Brain data set contains the same cases as the 2016 and 2017 Brain Tumor Segmentation (BraTS) challenges 36,37,38.

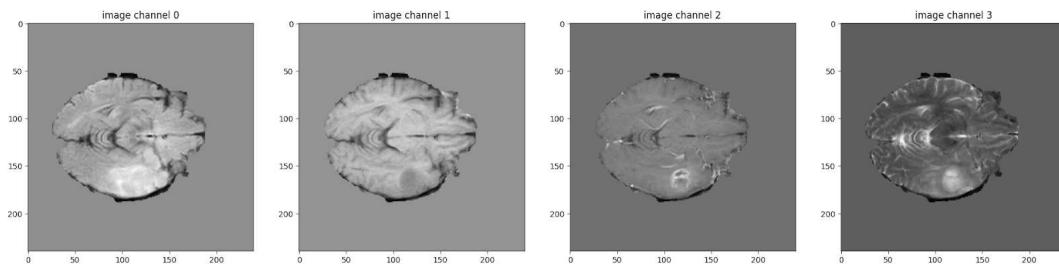
80% of the images are used for training and 20% are used for testing(validation) .

To quickly get started with popular training data, MONAI provides several ready-to-integrate Dataset classes (such as MedNISTDataset, DecathlonDataset, TciaDataset), which include data downloading, and support training/evaluation splits

generation with transforms. [Public datasets tutorial] The common workflow of predefined datasets:

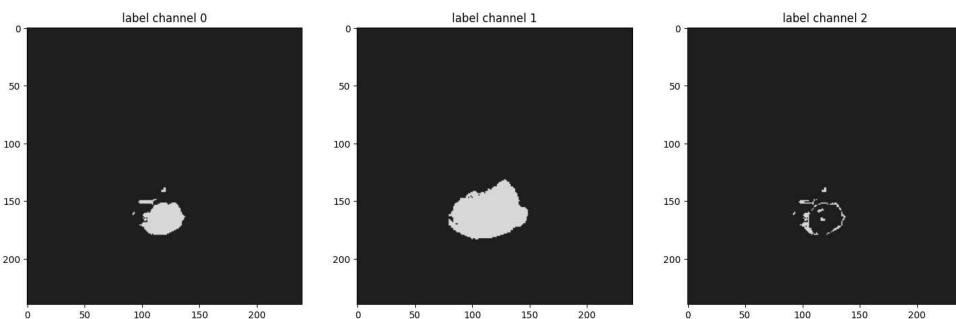


**Fig 6.1 The common workflow of predefined datasets**



## Fig 6.2 Brain MRI images

Monai provides us with a class called DecathlonDataset to automatically download the data of Medical Segmentation Decathlon challenge (<http://medicaldecathlon.com/>) and generate items for training, validation or test. It will also load these properties from the JSON config file of the dataset.



### **Fig 6.3 BRATS 3D segmentation**

### **6.1.2 Data Transforms**

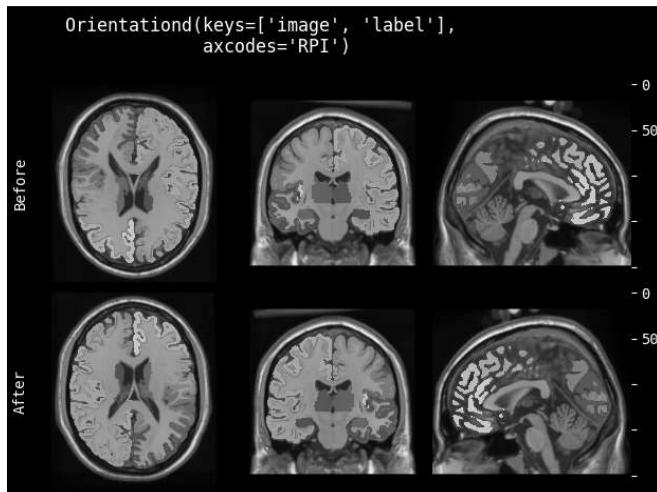
We have converted the labels into the BRATS class.

In the dataset, Label 1 represents peritumoral edema ,Label 2 represents GD-enhancing tumor and Label 3 represents the necrotic and non-enhancing tumor core.

The BRATS classes are TC(Tumor Core),WT(whole Tumor) and ET(Enhancing Tumor).We have merged label 2 and 3 to construct TC and merged label 1,2 and 3 to construct WT and label 2 is the ET.

We have written a class in python in order to do this.

Further with the help of monai transform we have loaded Nifti image with metadata, load a list of images and stack them.We augment the training images by changing the images orientation by using monai transform API called Orientationd.



**Fig 6.4 Segmentation of brain matter**

We have used the axcode as RAS- Right anterior Superior.

We have used RandSpatialCropd to crop the image randomly to the size of [224,224,144]

Further the image is flipped around our each axis to a random degree and Randomly

---

adjusted intensity for data augmentation.

We apply a random (per channel) intensity shift ( $-0.1..0.1$  of image std) and scale ( $0.9..1.1$ ) on input image channels. We also apply a random axis mirror flip (for all 3 axes) with a probability 0.5.

### **6.1.3 Modeling**

Convolutional neural networks (CNNs) are widely used in pattern and image recognition, including medical image analysis<sup>1</sup>. CNNs are promising candidates to handle medical image analysis because of their insensitivity to noise and the abilities to work with large input volumes and automatically extract features.

We have intended to train 2 models of architecture using the same dataset.

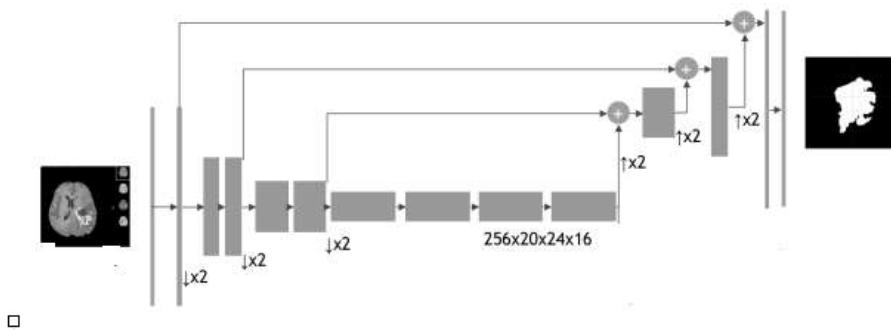
#### **6.1.3.1 SEGRESNET architecture**

SegResNet refers to a segmentation model based on the Residual Network (ResNet) architecture. ResNet is a popular deep convolutional neural network (CNN) architecture known for its ability to address the degradation problem in training very deep neural networks. It achieves this by using residual connections, which allow gradients to flow directly through the network, mitigating the vanishing gradient problem.

SegResNet is specifically designed for image segmentation tasks. It extends the ResNet architecture by incorporating skip connections, also known as skip connections or shortcut connections, to enable better feature propagation across different layers. These skip connections facilitate the fusion of low-level and high-level features, which is particularly important for segmentation tasks where precise localization is required.

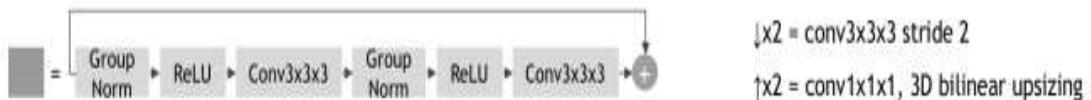
The skip connections in SegResNet typically involve concatenating or summing the feature maps from earlier layers with the feature maps from later layers. This allows the network to have access to both local and global contextual information, improving the accuracy and efficiency of segmentation.

By combining the powerful feature extraction capabilities of ResNet with the skip connections in SegResNet, the model can effectively capture both fine-grained details and high-level contextual information necessary for accurate image segmentation. This makes SegResNet a popular choice for various segmentation tasks, including medical image segmentation, object detection, and semantic segmentation in computer vision applications.



**Fig 6.5 Schematic visualization of the network architecture**

#### Architecture of each block



**Fig 6.6 Architecture of each block**

Our segmentation approach follows encoder-decoder based CNN architecture with an asymmetrically larger encoder to extract image features and a smaller decoder to reconstruct the segmentation mask.

#### Encoder part

The encoder part uses ResNet [10] blocks, where each block consists of two convolutions with normalization and ReLU, followed by additive identity skip connection. For normalization, we use Group Normalization (GN) [22], which shows better than BatchNorm performance when batch size is small (batch size

of 1 in our case). We follow a common CNN approach to progressively downsize image dimensions by 2 and simultaneously increase feature size by 2. For downsizing we use strided convolutions. All convolutions are 3x3x3 with initial number of filters equal to 32.

### **Decoder part**

The decoder structure is similar to the encoder one, but with a single block per each spatial level. Each decoder level begins with upsizing: reducing the number of features by a factor of 2 (using 1x1x1 convolutions) and doubling the spatial dimension (using 3D bilinear upsampling), followed by an addition of encoder output of the equivalent spatial level. The end of the decoder has the same spatial size as the original image, and the number of features equal to the initial input feature size, followed by 1x1x1 convolution into 3 channels and a sigmoid function.

#### **6.1.3.2 UNETR architecture**

UNETR is an abbreviation for "UNETR: Transformers for 3D Medical Image Segmentation." It refers to a framework proposed in a research paper that combines the U-Net architecture with transformers for 3D medical image segmentation.

Medical image segmentation plays a critical role in various clinical applications, such as tumor detection and organ delineation. Traditional approaches for image segmentation often rely on convolutional neural networks (CNNs). However, transformers, which have demonstrated exceptional performance in natural language processing tasks, offer a different perspective for capturing global context information.

In the UNETR framework, the authors replace the traditional U-Net encoder with a transformer encoder. U-Net is a widely used CNN-based architecture known for its success in medical image segmentation. However, it has limitations in capturing long-range dependencies in the image. Transformers, on the other hand, excel in capturing global context information through their self-attention mechanism.

---

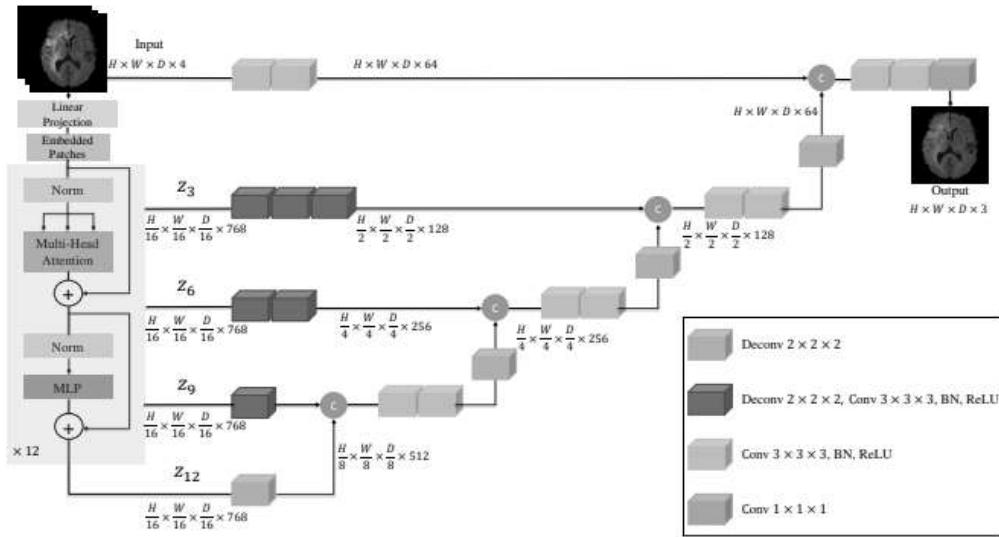
The transformer encoder in UNETR consists of multiple layers, each containing a multi-head self-attention mechanism and position-wise feed-forward networks. The transformer encoder captures long-range dependencies and extracts rich contextual information from the 3D medical images. The output of the transformer encoder is then passed to the decoder, which follows the U-Net architecture. The decoder progressively up-samples the features and combines them with the corresponding features from the encoder to produce the final segmentation map.

The authors train UNETR using a combination of supervised and self-supervised learning. They utilize a large-scale dataset of 3D medical images with manually annotated ground truth segmentation maps for supervised training. Additionally, they propose a self-supervised pre-training step using a contrastive loss, where the model learns to distinguish between positive and negative pairs of augmented images.

Experimental results demonstrate that UNETR outperforms traditional CNN-based approaches for 3D medical image segmentation. It achieves state-of-the-art performance on benchmark datasets, including the BraTS dataset for brain tumor segmentation. The transformer encoder's ability to capture long-range dependencies contributes to the improved performance.

In summary, UNETR is a framework that combines transformers with the U-Net architecture for 3D medical image segmentation. By leveraging the self-attention mechanism of transformers, UNETR captures global context information and achieves superior performance compared to traditional CNN-based methods. It holds promise for enhancing various clinical applications by enabling more accurate and efficient segmentation of 3D medical images.

In figure UNETR utilizes a contracting-expanding pattern consisting of a stack of transformers as the encoder which is connected to a decoder via skip connections. As commonly used in NLP, the transformers operate on 1D sequence of input embeddings. Similarly, 1D sequence of a 3D input volume  $x \in \mathbb{R}^{H \times W \times D \times C}$  with resolution  $(H, W, D)$  and  $C$  input channels by dividing it into flattened uniform non-overlapping patches  $x_v \in \mathbb{R}^{N \times (P^3 \cdot C)}$  where  $(P, P, P)$  denotes the resolution of each patch and  $N = (H \times W \times D) / P^3$  is the length of the sequence.



**Fig 6.7 Overview of the UNETR architecture**

#### 6.1.4 TRAINING

##### Loss Function

$L_{dice}$  is a soft dice loss [19] applied to the decoder output  $p_{pred}$  to match the segmentation mask  $p_{true}$ .

$$L_{dice} = \frac{2 * \sum p_{true} * p_{pred}}{\sum p_{true}^2 + \sum p_{pred}^2 + \epsilon}$$

**Fig 6.8 Loss function**

with smooth\_dr=1e-5

##### Optimizer Function

We use Adam optimizer with initial learning rate of  $\alpha_0 = 1e-4$  and progressively decrease it according to:

$$\alpha = \alpha_0 * \left(1 - \frac{e}{N_e}\right)^{0.9}$$

**Fig 6.9 Optimizer function**

where e is an epoch counter, and Ne is a total number of epochs

#### **6.1.4.1 Ensemble**

Results of each image in the validation data is given as input to both the models. The result of both models is combined by stacking the results and taking the mean of the tensor at zero dimensionality.

This is done in order to benefit from the learnings of both the models and achieve a better dice score for the validation data.

```
def __call__(self, img: Union[Sequence[NdarrayOrTensor], NdarrayOrTensor]):
    img_ = self.get_stacked_torch(img)
    if self.weights is not None:
        self.weights = self.weights.to(img_.device)
        shape = tuple(self.weights.shape)
        for _ in range(img_.ndimension() - self.weights.ndimension()):
            shape += (1,)
        weights = self.weights.reshape(*shape)

        img_ = img_ * weights / weights.mean(dim=0, keepdim=True)

    out_pt = torch.mean(img_, dim=0)
    return self.post_convert(out_pt, img)
```

**Fig 6.10 Ensemble**

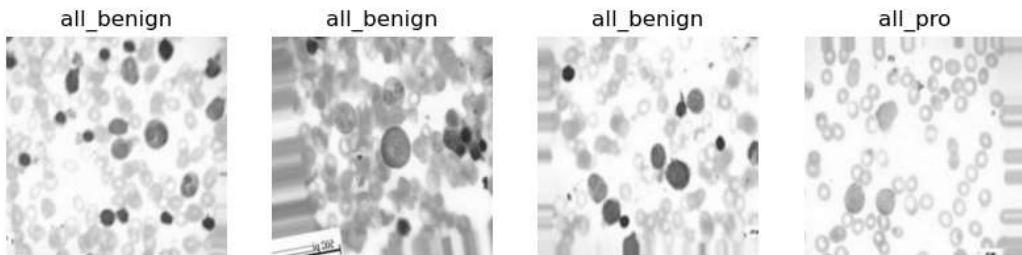
## **6.2 Implementation of Classification Models**

We have trained 4 models which classify different types of Cancer. The trained models are DenseNet201, MobileNetV3Small, VGG19 and ResNet50V2.

### **6.2.1 Classification of different types of Cancer**

#### **1. Acute Lymphoblastic Leukemia**

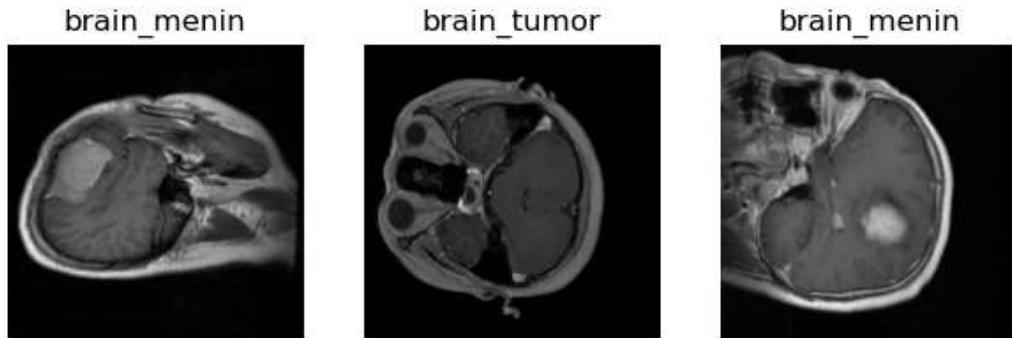
- a. Benign
- b. Early
- c. Pre
- d. Pro



**Fig 6.11 Acute Lymphoblastic Leukemia**

**2. Brain Cancer**

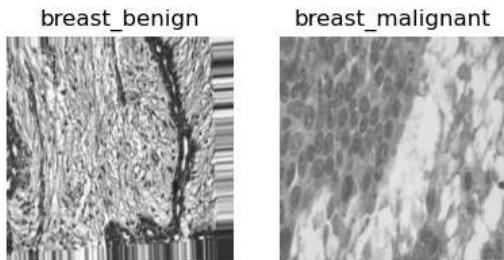
- a. Glioma
- b. Meningioma
- c. Pituitary



**Fig 6.12 Brain Cancer**

**3. Breast Cancer**

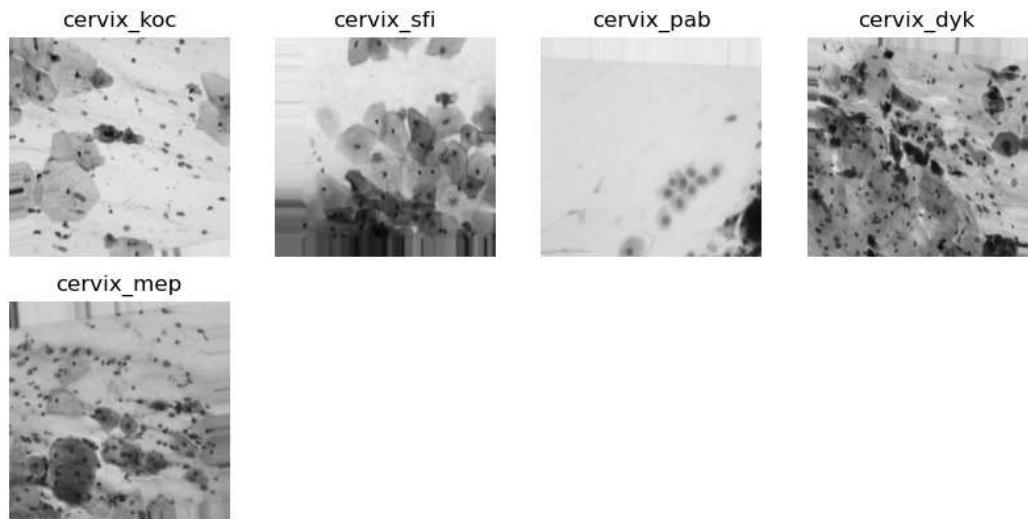
- a. Benign
- b. Malignant



**Fig 6.13 Breast Cancer**

**4. Cervical Cancer**

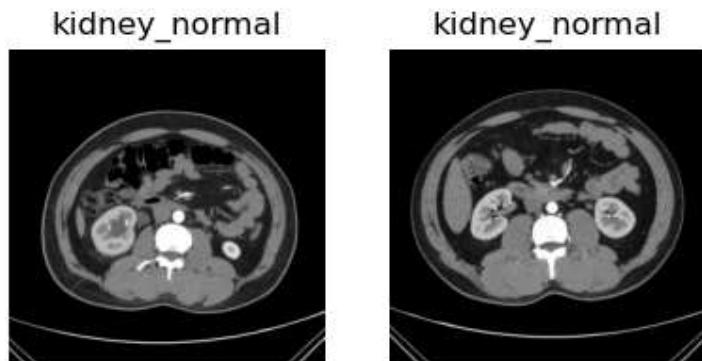
- a. Dyskeratotic
- b. Koilocytotic
- c. Metaplastic
- d. Parabasal
- e. Superficial-Intermediate



**Fig 6.14 Cervical Cancer**

**5. Kidney Cancer**

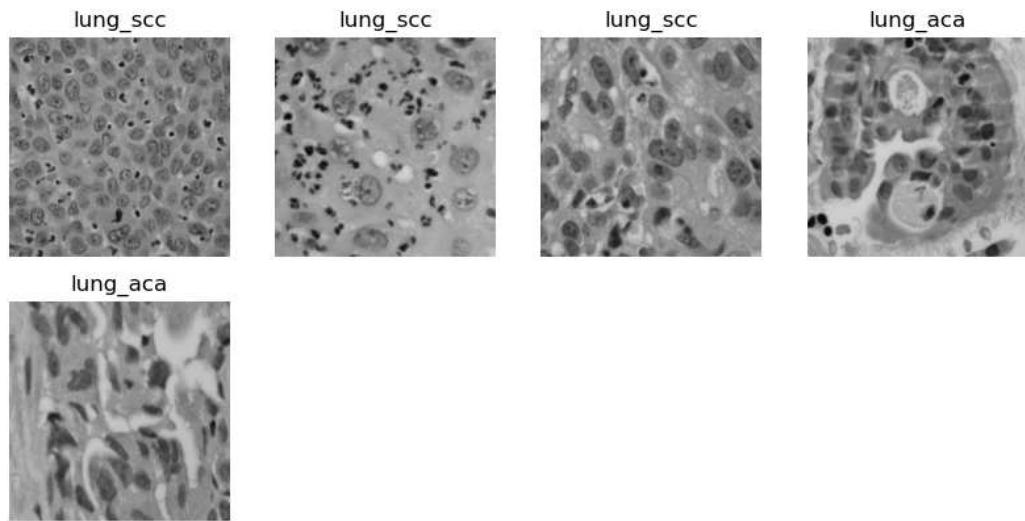
- a. Normal
- b. Tumor



**Fig 6.15 Kidney Cancer**

**6. Lung and Colon Cancer**

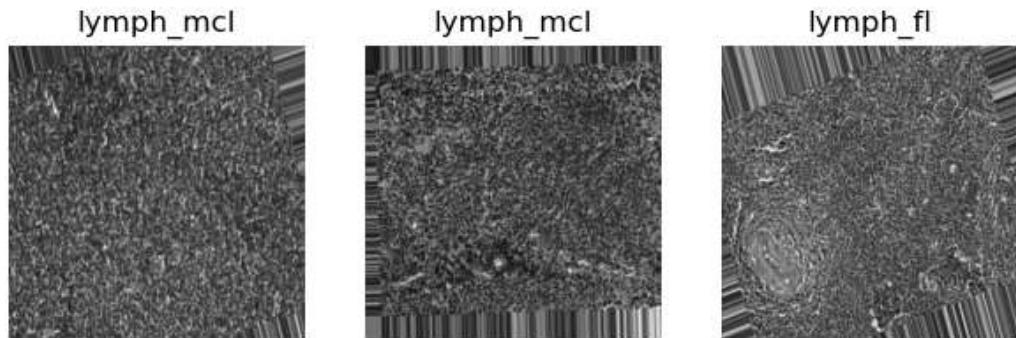
- a. Colon Adenocarcinoma
- b. Colon Benign Tissue
- c. Lung Adenocarcinoma
- d. Lung Benign Tissue
- e. Lung Squamous Cell



**Fig 6.16 Lung Cancer**

### **7. Lymphoma**

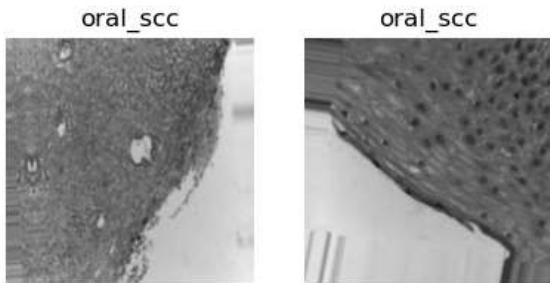
- a. Chronic Lymphocytic Leukemia
- b. Follicular Lymphoma
- c. Mantle Cell



**Fig 6.17 Lymphoma**

### **8. Oral Cancer**

- a. Normal
- b. Oral Squamous cell



**Fig 6.18 Oral Cancer**

### **6.2.2 DenseNet201 Architecture**

DenseNet, short for Dense Convolutional Network, is a deep learning architecture that was introduced by Huang et al. in 2016. It is known for its unique connectivity pattern and has achieved state-of-the-art performance on several image classification tasks.

In DenseNet, the basic building block is called a "dense block." A dense block consists of several convolutional layers that are densely connected to each other. Let's consider a dense block with  $L$  layers. The output of the  $l$ -th layer is denoted as  $H(l)$ , where  $l$  ranges from 1 to  $L$ .

The input to the first layer,  $H(0)$ , is the input image or the output of the preceding layer. Each layer within the dense block takes all preceding feature maps as inputs and produces its own feature maps as outputs. In other words,  $H(l)$  is obtained by concatenating the feature maps from all previous layers:  $H(l) = [H(0), H(1), \dots, H(l-1)]$ .

To maintain the same spatial dimensions throughout the dense block, the convolutional layers typically use padding. Batch normalization and a non-linear activation function, such as ReLU, are applied after each convolution. This ensures the normalization and non-linearity of the features.

DenseNet-201, also known as DenseNet-201, is a deep convolutional neural network (CNN) architecture that belongs to the DenseNet family. It was introduced by Huang et al. in their paper "Densely Connected Convolutional Networks" in 2017.

DenseNet is designed to address the issue of vanishing gradients and strengthen feature propagation in deep networks. It achieves this by introducing dense

---

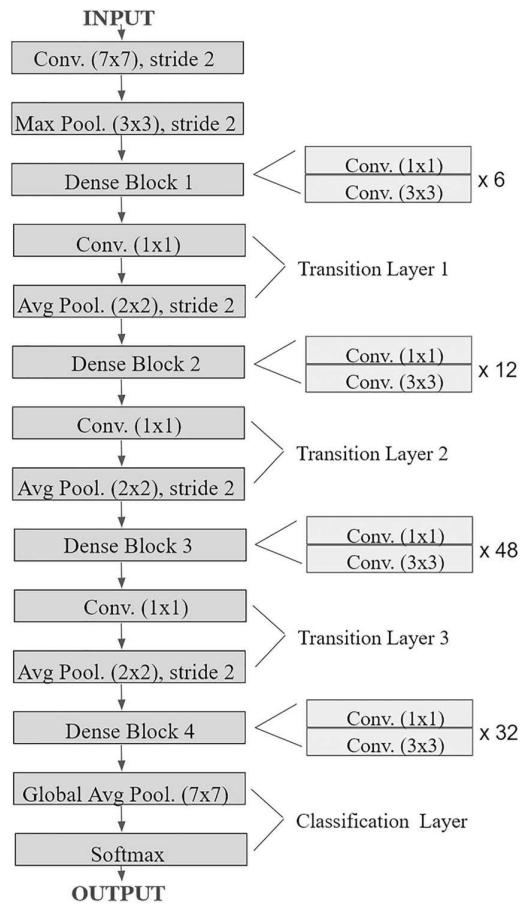
connections, which connect each layer to every other layer in a feed-forward fashion. This dense connectivity facilitates the direct flow of gradients and encourages feature reuse across different layers, enabling better information flow and improved gradient flow during training.

DenseNet-201 specifically refers to a variant of DenseNet that consists of 201 layers. It is a considerably deep network architecture and is known for its powerful feature extraction capabilities. The architecture of DenseNet-201 includes dense blocks, transition layers, and a global average pooling layer followed by a fully connected layer for classification.

In the dense blocks, each layer receives the feature maps from all preceding layers in a densely connected manner. This dense connectivity enables rich feature representations and encourages feature reuse, resulting in compact and efficient models. The transition layers are used to downsample the feature maps and reduce the dimensionality, which helps control computational complexity.

DenseNet-201 has been widely used in various computer vision tasks, including image classification, object detection, and segmentation. With its deep architecture and dense connections, it has demonstrated strong performance on benchmark datasets, often achieving state-of-the-art results.

In summary, DenseNet-201 is a deep convolutional neural network architecture that employs dense connections to address the vanishing gradient problem and enhance feature propagation. With its 201 layers, it provides a powerful framework for various computer vision tasks, leveraging the benefits of dense connectivity for efficient and accurate feature extraction.



**Fig 6.19 DenseNet Architecture**

### 6.2.3 MobileNetV3Small architecture

MobileNet is a convolutional neural network architecture specifically designed for mobile and embedded devices with limited computational resources. It was introduced by Howard et al. in 2017 as a lightweight and efficient alternative to traditional deep learning models.

The main objective of MobileNet is to strike a balance between model size and accuracy, making it suitable for deployment on devices with low memory, limited processing power, and restricted energy consumption.

MobileNet achieves its efficiency by utilizing depth-wise separable convolutions. Traditional convolutions perform both spatial filtering (convolving across the width and height dimensions) and cross-channel filtering (convolving across the channel

---

dimension) simultaneously. In contrast, depth-wise separable convolutions split these two operations into separate layers.

MobileNetV3Small refers to a variant of the MobileNetV3 architecture, which is a family of lightweight convolutional neural network (CNN) models specifically designed for mobile and embedded devices. MobileNetV3Small is a smaller version of MobileNetV3 that aims to strike a balance between model size, computational efficiency, and accuracy.

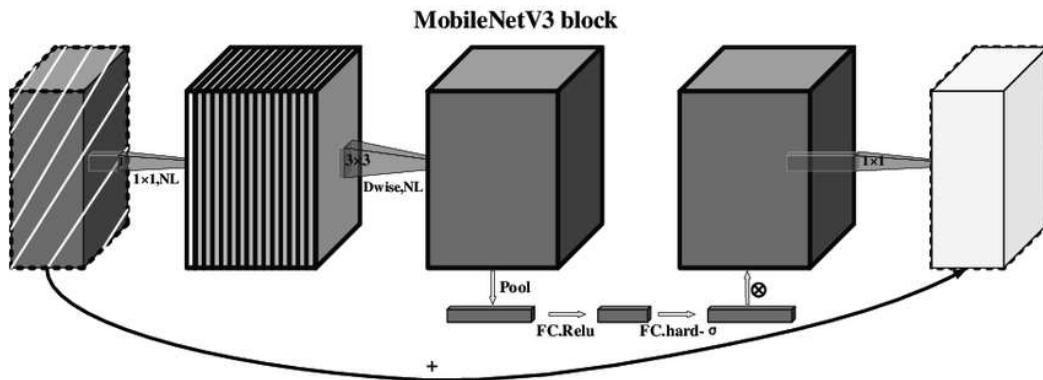
The MobileNetV3 architecture introduces several key design elements to improve performance over previous iterations, such as MobileNetV1 and MobileNetV2. These elements include:

- Inverted Residuals: MobileNetV3 utilizes inverted residuals, which involve using a lightweight linear bottleneck layer followed by a more computationally expensive expansion layer. This configuration reduces the computational cost while preserving important features.
- MobileNetV3 Blocks: The network is composed of a series of blocks, each containing a combination of depthwise separable convolutions, activation functions, and pointwise convolutions. These blocks are stacked together to form the overall architecture.
- Efficient Use of Activations: MobileNetV3 employs advanced activation functions like Hard Swish, which strikes a balance between non-linearity and efficiency by using a piecewise linear approximation of the Swish activation function.

MobileNetV3Small is a compact version of the MobileNetV3 architecture, designed to have a smaller model size and lower computational requirements while still maintaining reasonable accuracy. The exact specifications of MobileNetV3Small may vary, but it typically consists of fewer layers and parameters compared to the full MobileNetV3 model.

MobileNetV3Small is suitable for resource-constrained scenarios where computational resources, memory, or power consumption are limited, such as mobile

devices, edge computing, or real-time applications. It offers a trade-off between model size, speed, and accuracy, making it well-suited for applications that require lightweight and efficient deep learning models.



**Fig 6.20 MobileNet Architecture**

### 6.2.3 ResNet50v2 Architecture

ResNet50v2 refers to the Residual Network (ResNet) architecture with 50 layers, and it is an improved version of the original ResNet50 architecture. ResNet50v2 was introduced by Kaiming He et al. in their paper "Identity Mappings in Deep Residual Networks" in 2016.

The ResNet architecture revolutionized deep learning by addressing the problem of training very deep neural networks. It introduced residual connections, also known as skip connections or shortcut connections, to enable the flow of gradients and address the vanishing gradient problem. These connections allow the network to learn residual mappings, i.e., the difference between the input and the desired output.

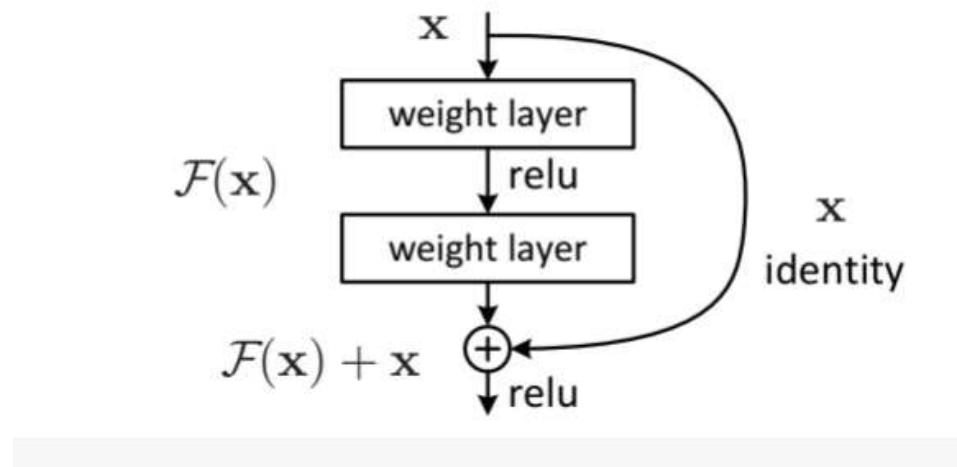
ResNet50v2 builds upon the original ResNet50 architecture and incorporates some enhancements to further improve its performance. Some key features of ResNet50v2 include:

- Identity Mapping: ResNet50v2 promotes the use of identity mappings by introducing an "identity" shortcut connection. This connection bypasses the layers and preserves the input features, allowing the network to learn an identity mapping when it is optimal. This helps alleviate the degradation problem that can occur when stacking more layers.
- Bottleneck Structure: ResNet50v2 employs a bottleneck structure in its residual blocks. This structure consists of three layers: 1x1 convolutional layer (to reduce dimensionality), 3x3 convolutional layer (to capture spatial information), and another 1x1 convolutional layer (to restore dimensionality). The bottleneck structure reduces computational complexity while maintaining representational capacity.
- Pre-Activation: ResNet50v2 introduces pre-activation units, where the batch normalization and activation functions are applied before convolution operations. This ordering helps in mitigating the vanishing gradient problem and improves information flow during training.
- Deeper Architecture: ResNet50v2 has 50 layers, making it deeper than the original ResNet50. The additional layers allow the network to learn more complex and abstract features, potentially improving its performance on challenging tasks.

ResNet50v2 has been widely adopted in various computer vision tasks, including image classification, object detection, and semantic segmentation. It has demonstrated strong performance and is often used as a baseline architecture for benchmarking and comparison.

In summary, ResNet50v2 is an enhanced version of the ResNet50 architecture. It incorporates identity mappings, bottleneck structures, pre-activation, and a deeper architecture to improve training and performance. ResNet50v2 is a powerful and widely used architecture in computer vision applications, particularly in tasks that require deep and accurate feature extraction.

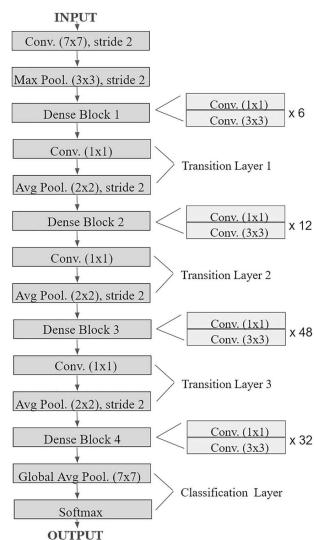
### **Residual blocks: The building blocks of ResNet**



**Fig 6.21 Example of Residual Block**

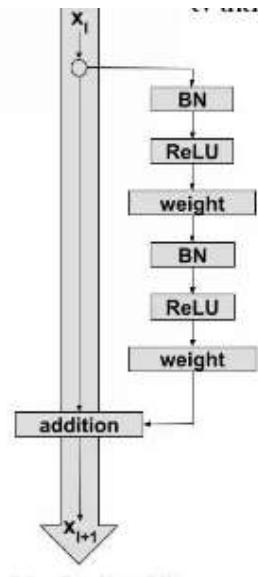
**Residual blocks** are the building blocks of ResNet. Let us look at these residual blocks.

We can understand the structure of the residual blocks from the above image. As we can see, there is a direct connection between some layers, which skips the in-between layers. This is called skip connection and is the core of residual blocks. Due to these skip connections, the output is changed.



**Fig 6.22 Residual Blocks**

Without these skip connections, the input ( $X$ ) is passed through all the layers in between and is multiplied by the corresponding weights of the layer and then added by a bias term. Then it goes through an activation function to give us the final outcome.



**Fig 6.23 RESNETv2**

ResNet-V1 adds a second non-linearity after the addition of  $F(X)$  and  $X$ . ResNet-V2, on the other hand, removed this non-linearity.

ResNet-V1 performs convolution after the batch normalization and ReLU activation. ResNet-V2 uses Batch Normalization and ReLU activation to the input before its multiplication with the weight( $W$ ) matrix.

### 6.2.5 VGG19 Architecture

VGG19 refers to the 19-layer variant of the VGG (Visual Geometry Group) architecture, which was proposed by Simonyan and Zisserman in their paper "Very Deep Convolutional Networks for Large-Scale Image Recognition" in 2014.

The VGG architecture gained popularity for its simplicity and effectiveness in image classification tasks. VGG19, being one of its deeper variants, is composed of 19 layers, including convolutional layers, max pooling layers, and fully connected layers.

---

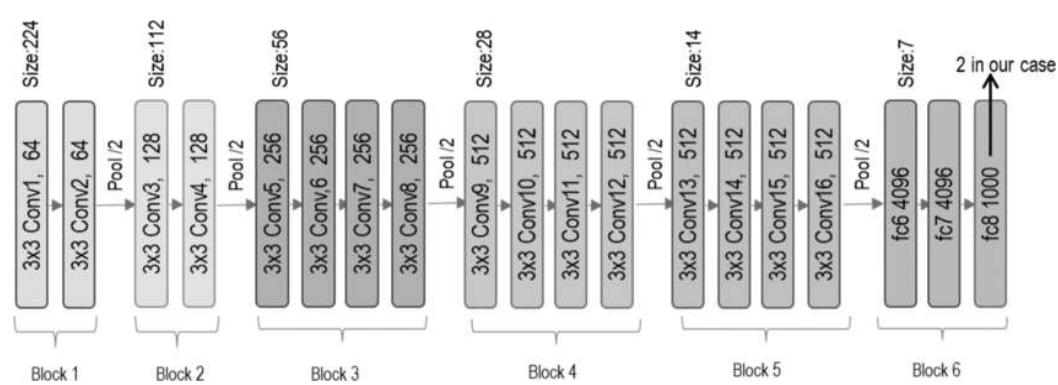
Key characteristics of the VGG19 architecture include:

- Convolutional Layers: VGG19 consists of a series of convolutional layers, each followed by a rectified linear unit (ReLU) activation function. These layers are responsible for capturing hierarchical features from the input image.
- Max Pooling Layers: After a set of convolutional layers, VGG19 includes max pooling layers, which reduce the spatial dimensions of the feature maps while retaining the most salient information. Max pooling helps create translation invariance and reduces the computational cost.
- Fully Connected Layers: Towards the end of the architecture, VGG19 has fully connected layers, which combine the learned features and perform the final classification. These layers are typically followed by a softmax activation function to produce class probabilities.
- Increasing Depth: VGG19 is characterized by its deep structure with 19 layers. The depth of the architecture allows for more expressive power in capturing intricate patterns and complex representations from the input data.

VGG19 was primarily designed for image classification tasks, particularly on large-scale datasets such as ImageNet. By stacking multiple layers and using relatively small convolutional filters (3x3), the architecture can capture both local and global information from the images, leading to better recognition accuracy.

Although VGG19 achieved remarkable results in image classification, it has a high number of parameters, making it computationally expensive and memory-intensive. However, the architecture has served as a valuable baseline and reference for subsequent deep learning models.

In summary, VGG19 is a deep convolutional neural network architecture with 19 layers, consisting of convolutional, max pooling, and fully connected layers. It has been influential in the field of image recognition and has played a significant role in advancing deep learning research.



**Fig 6.24 VGG19 Architecture**

VGG-19 has 16 convolution layers grouped into 5 blocks. After every block, there is a Maxpool layer that decreases the size of the input image by 2 and increases the number of filters of the convolution layer also by 2. The dimensions of the last three dense layers in block 6 are 4096, 4096, and 1000 respectively.

## 6.2.6 Training

### Optimizer

The Adam optimizer is an adaptive optimization algorithm commonly used in deep learning. It is known for its efficiency, effectiveness, and robustness in optimizing neural networks.

The key idea behind the Adam optimizer is to combine the benefits of two other popular optimization methods: Adaptive Moment Estimation (Adam) and Root Mean Square Propagation (RMSprop). By incorporating elements from both algorithms, Adam offers adaptive learning rates and momentum-based updates.

Here are the main components and characteristics of the Adam optimizer:

**Adaptive Learning Rates:** Adam adapts the learning rate for each parameter during training. It maintains a separate learning rate per parameter based on their past gradients. This adaptivity helps in speeding up convergence and handling different learning rates for different parameters.

---

Momentum Optimization: Adam utilizes the concept of momentum, which allows the optimizer to accumulate velocity or momentum over time. This momentum helps in navigating through flat regions and accelerating the learning process, especially in the presence of sparse gradients.

Bias Correction: To account for the initialization bias and the effect of moment estimates in the early stages of training, Adam performs a bias correction by scaling the initial estimates of the first and second moments.

Weight Decay: Adam includes an optional weight decay term, which helps prevent overfitting by adding a penalty to the objective function based on the magnitudes of the weights.

Computational Efficiency: Adam leverages vectorized operations and parallelism, making it computationally efficient and well-suited for large-scale deep learning models.

The Adam optimizer is widely used in various deep learning tasks, including image classification, object detection, natural language processing, and more. It has become a popular choice due to its fast convergence, robustness to different types of neural network architectures, and adaptive learning rate capabilities.

In TensorFlow's Keras API, you can create an Adam optimizer by specifying the learning rate as a parameter, as shown in the code snippet `tf.keras.optimizers.Adam(learning_rate=lr)`. This allows you to customize the learning rate value to optimize the training process for your specific task and model.

We have set the learning rate to 0.01.

## **Annealer**

ReduceLROnPlateau is a learning rate scheduling technique commonly used during training in deep learning models. It is an approach to dynamically adjust the learning rate based on the performance of the model on a validation set or a specific metric.

---

The purpose of ReduceLROnPlateau is to automatically reduce the learning rate when the model's performance plateaus or reaches a stagnant phase. This adjustment can help the model to converge better or find a more optimal solution. It is particularly useful when the learning rate needs to be fine-tuned during training to overcome local minima or speed up convergence.

Here are the key components and functionalities of ReduceLROnPlateau:

**Monitor Metric:** You specify a metric (e.g., validation loss or accuracy) to monitor during training. The learning rate adjustment is triggered based on the behavior of this metric.

**Patience:** The patience parameter determines the number of epochs to wait before reducing the learning rate if no improvement is observed. If the monitored metric does not improve for a specified number of epochs, the learning rate is adjusted.

**Factor:** The factor parameter determines the factor by which the learning rate is reduced. For example, if the factor is set to 0.1, the learning rate will be multiplied by 0.1 when the adjustment is triggered.

**Mode:** The mode parameter specifies whether to monitor the metric for improvement (e.g., "min" for loss) or for a plateau (e.g., "max" for accuracy). The mode determines the direction of improvement that is considered significant.

**Threshold:** The threshold parameter defines the threshold for the metric to be considered as a plateau. If the metric does not improve by more than the specified threshold, it is considered a plateau.

**Cool-down:** The cool-down parameter determines the number of epochs to wait after the learning rate is adjusted before resuming the regular training process. This cooldown period helps prevent immediate subsequent adjustments.

By using ReduceLROnPlateau, the learning rate can be adjusted dynamically during training, potentially improving the model's performance and convergence. It helps to find an appropriate learning rate schedule without manual tuning.

---

In TensorFlow and Keras, you can incorporate ReduceLROnPlateau by using the `tf.keras.callbacks.ReduceLROnPlateau` callback. This callback is passed to the `fit` method during training and is triggered based on the specified criteria to adjust the learning rate accordingly.

We have used “`ReduceLROnPlateau(monitor='val_accuracy', factor=0.5, patience=5, verbose=1, min_lr=1e-3)`”

### **6.2.7 Ensemble**

Ensemble learning is a machine learning technique that involves combining multiple individual models, called base models or weak learners, to create a more robust and accurate prediction model. The idea behind ensemble learning is that by combining the predictions of multiple models, the ensemble can outperform any individual model in terms of accuracy, generalization, and robustness.

An ensemble can be created using various techniques, including:

**Voting:** In this approach, each base model in the ensemble independently predicts the target variable, and the final prediction is determined based on majority voting (for classification problems) or averaging (for regression problems) of the individual predictions.

**Bagging:** Bagging stands for bootstrap aggregating. It involves creating multiple subsets of the training data through bootstrap sampling and training a separate base model on each subset. The final prediction is obtained by aggregating the predictions of all base models, typically through voting or averaging.

**Boosting:** Boosting is an iterative ensemble technique where each base model is trained sequentially, with each subsequent model focusing on correcting the mistakes made by the previous models. The final prediction is a weighted combination of all the base models' predictions.

**Stacking:** Stacking involves training multiple base models on the original training data and using their predictions as input features for a higher-level model, called a

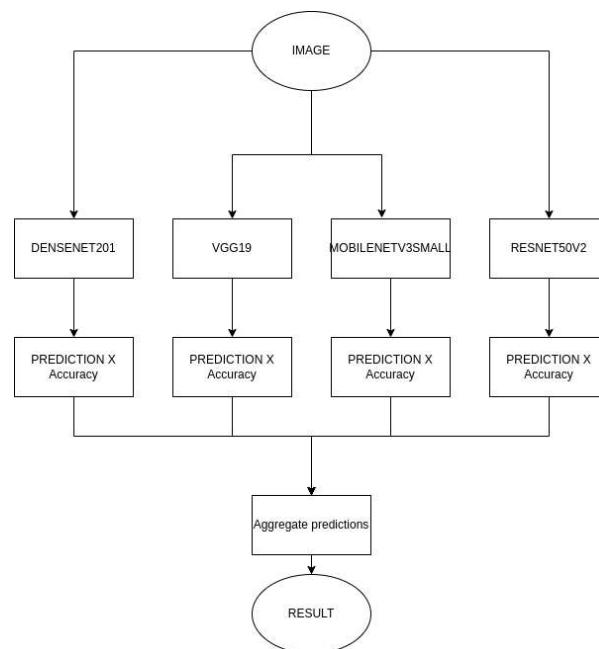
meta-learner. The meta-learner learns to combine the base models' predictions to make the final prediction.

Ensemble learning can provide several benefits, including improved accuracy, reduced overfitting, better generalization to unseen data, and increased robustness to noise and outliers. It is widely used in various machine learning tasks, such as classification, regression, and anomaly detection.

Ensemble methods can be implemented using a variety of algorithms, such as Random Forest, Gradient Boosting Machines (GBM), AdaBoost, and Stacking. The choice of ensemble method depends on the problem domain, dataset characteristics, and specific requirements.

Overall, ensemble learning is a powerful technique that leverages the strengths of multiple models to create a more accurate and reliable prediction model, and it is commonly used to achieve state-of-the-art performance in many machine learning applications.

We have tried to ensemble the predictions of all the models based on the accuracy as a heuristic knowledge to weight the predictions of each model.



**Fig 6.25 Ensemble Technique Overview**

## CHAPTER 7

# PERFORMANCE METRICS

## 7.1 Metrics Used for Segmentation

### Dice Score

The Dice score, also known as the Dice coefficient or Dice similarity coefficient, is a metric commonly used to evaluate the performance of image segmentation algorithms. It measures the overlap between the predicted segmentation mask and the ground truth mask.

The Dice score is computed as the ratio of twice the intersection of the predicted and ground truth masks to the sum of the sizes of the predicted and ground truth masks. Mathematically, it can be expressed as:

$$\text{Dice Score} = (2 * \text{Intersection}) / (\text{Size of Predicted Mask} + \text{Size of Ground Truth Mask})$$

The Intersection refers to the number of pixels or voxels where the predicted and ground truth masks both have a positive value. The Size of Predicted Mask and Size of Ground Truth Mask correspond to the total number of pixels or voxels that are positive in each mask, respectively.

The Dice score ranges from 0 to 1, with 1 indicating a perfect overlap between the predicted and ground truth masks (i.e., a perfect segmentation), and 0 indicating no overlap (i.e., complete mismatch).

The Dice score is widely used in medical image segmentation tasks, such as tumor segmentation, where it provides a measure of how well the algorithm captures the target structure. It is particularly useful when dealing with imbalanced classes or when the focus is on the overlapping region rather than the entire image. Higher

---

Dice scores indicate better segmentation performance, with values closer to 1 indicating more accurate and precise segmentations.

## 7.2 Metrics Used for Classification

### Performance Metrics

After doing the usual Feature Engineering, Selection, and of course, implementing a model and getting some output in forms of a probability or a class, the next step is to find out how effective is the model based on some metric using test datasets. Different performance metrics are used to evaluate different Machine Learning Algorithms. For now, we will be focusing on the ones used for Classification problems. We can use classification performance metrics such as LogLoss, Accuracy, AUC(Area under Curve) etc. Another example of metric for evaluation of machine learning algorithms is precision, recall, which can be used for sorting algorithms primarily used by search engines.

### Confusion Matrix

The Confusion matrix is one of the most intuitive and easiest metrics used for finding the correctness and accuracy of the model. It is used for Classification problems where the output can be of two or more types of classes. Before diving into what the confusion matrix is all about and what it conveys, Let's say we are solving a classification problem where we are predicting whether a person is having cancer or not. Let's give a label to our target variable: 1: When a person is having cancer 0: When a person is NOT having cancer. Alright! Now that we have identified the problem, the confusion matrix is a table with two dimensions (“Actual” and “Predicted”), and sets of “classes” in both dimensions. Our Actual classifications are columns Testing 37 and Predicted ones are Rows. The Confusion matrix in itself is not a performance measure as such, but almost all of the performance metrics are based on Confusion Matrix and the numbers inside it.

		Actual Values	
		Positive	Negative
Predicted Values	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

Fig 7.1 Confusion Matrix

Terms associated with Confusion matrix:

1. **True Positives (TP):** True positives are the cases when the actual class of the data point was 1(True) and the predicted is also 1(True) Ex: The case where a person is actually having cancer(1) and the model classifying his case as cancer(1) comes under True positive.
2. **True Negatives (TN):** True negatives are the cases when the actual class of the data point was 0(False) and the predicted is also 0(False) Ex: The case where a person NOT having cancer and the model classifying his case as Not cancer comes under True Negatives. Testing 38
3. **False Positives (FP):** False positives are the cases when the actual class of the data point was 0(False) and the predicted is 1(True). False is because the model has predicted incorrectly and positive because the class predicted was a positive one. (1) Ex: A person NOT having cancer and the model classifying his case as cancer comes under Positives.

4. **False Negatives (FN):** False negatives are the cases when the actual class of the data point was 1(True) and the predicted is 0(False). False is because the model has predicted incorrectly and negative because the class predicted was a negative one. (0) Ex: A person having cancer and the model classifying his case as No-cancer comes under False Negatives. The ideal scenario that we all want is that the model should give 0 False Positives and 0 False Negatives. But that's not the case in real life as any model will NOT be 100% accurate most of the time.

### Accuracy

Accuracy in classification problems is the number of correct predictions made by the model over all kinds of predictions made. In the Numerator, are our correct predictions (True positives and True Negatives) (Marked as red in the fig above) and in the denominator, are the kind of all predictions made by the algorithm (Right as well as wrong ones). Accuracy is a good measure when the target variable classes in the data are nearly balanced. Fig 7.4 shows accuracy derived from the confusion matrix.

$$\text{Accuracy} = \frac{\text{TrueNegatives} + \text{TruePositive}}{\text{TruePositive} + \text{FalsePositive} + \text{TrueNegative} + \text{FalseNegative}}$$

**Fig 7.2 Accuracy**

### Precision

Precision is defined as the ratio of correctly classified positive samples (True Positive) to a total number of classified positive samples (either correctly or incorrectly). Fig 7.4 shows precision derived from the confusion matrix. Precision = True Positive/True Positive + False Positive Precision = TP/TP+FP TP- True Positive FP- False Positive The precision of a machine learning model will be low when the value of; TP+FP (denominator) > TP (Numerator) The precision of the

machine learning model will be high when Value of; TP (Numerator) > TP+FP (denominator) Hence, precision helps us to visualize the reliability of the machine learning model in classifying the model as positive.

$$\text{Precision} = \frac{\text{True Positive}(TP)}{\text{True Positive}(TP) + \text{False Positive}(FP)}$$

**Fig 7.3 Precision**

### Recall or Sensitivity

The recall is calculated as the ratio between the numbers of Positive samples correctly classified as Positive to the total number of Positive samples. The recall measures the model's ability to detect positive samples. The higher the recall, the more positive samples detected. Fig 7.4 shows recall derived from confusion matrix. Recall = True Positive/True Positive + False Negative Testing 40 Recall = TP/TP+FN TP- True Positive FN- False Negative Recall of a machine learning model will be low when the value of; TP+FN (denominator) > TP (Numerator) Recall of machine learning model will be high when Value of; TP (Numerator) > TP+FN (denominator) Unlike Precision, Recall is independent of the number of negative sample classifications. Further, if the model classifies all positive samples as positive, then Recall will be 1.

$$\text{Recall} = \frac{\text{True Positive}(TP)}{\text{True Positive}(TP) + \text{False Negative}(FN)}$$

**Fig 7.4 Recall**

### F1 score

The F1 score is defined as the harmonic mean of precision and recall. In the F1 score, we compute the average of precision and recall. They are both rates, which

makes it a logical choice to use the harmonic mean. The F1 score formula is shown here: This makes that the formula for the F1 score is the following: Since the F1 score is an average of Precision and Recall, it means that the F1 score Testing 41 gives equal weight to Precision and Recall: A model will obtain a high F1 score if both Precision and Recall are high A model will obtain a low F1 score if both Precision and Recall are low A model will obtain a medium F1 score if one of Precision and Recall is low and the other is high.

$$F1\ Score = 2 \times \frac{recall \times precision}{recall + precision}$$

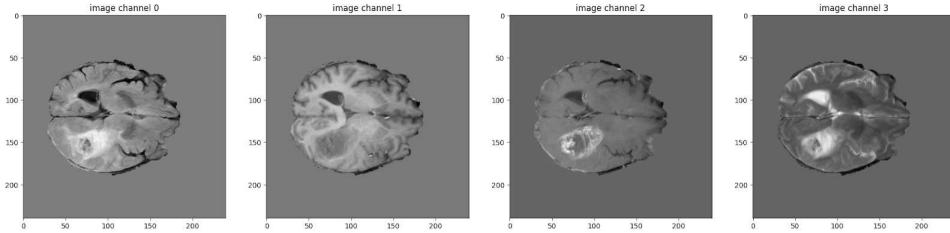
**Fig 7.5 F1 Score**

# CHAPTER 8

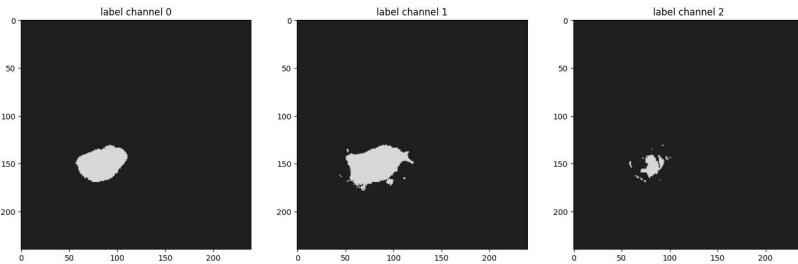
## PERFORMANCE ANALYSIS

### 8.1 RESULT FOR BRAIN TUMOR SEGMENTATION

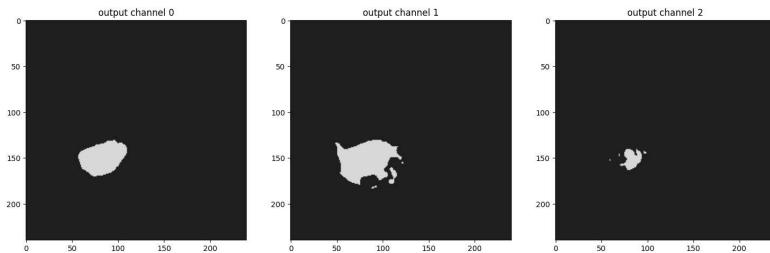
#### SEGRESNET



**Fig 8.1 Original Image**



**Fig 8.2 SEGRESNET - Ground Truth**



**Fig 8.3 SEGRESNET - Output**

#### Dice Scores:

Metric on original image spacing: 0.7625482082366943

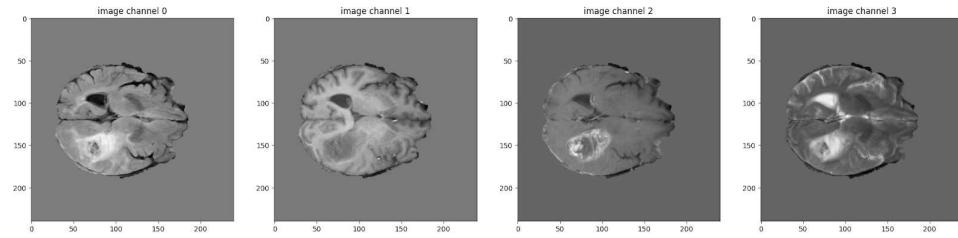
metric\_tc: 0.8078

metric\_wt: 0.9002

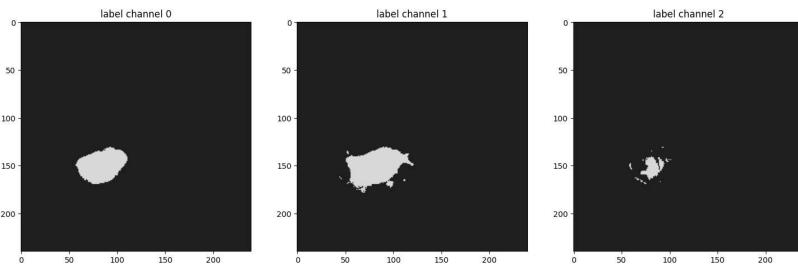
metric\_et: 0.5796

Results achieved at 63 epochs

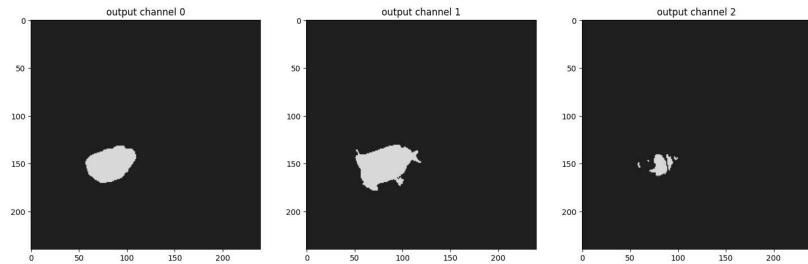
## **UNETR**



**Fig 8.4 Original Image**



**Fig 8.5 UNETR - Ground Truth**



**Fig 8.6 UNETR - Output**

### **Dice Scores:**

Metric on original image spacing: **0.737675666809082**

**metric\_tc: 0.7786**

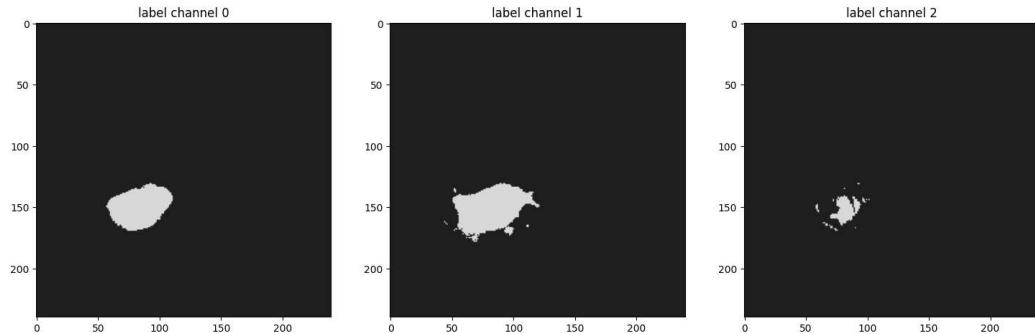
**metric\_wt: 0.8912**

**metric\_et: 0.5432**

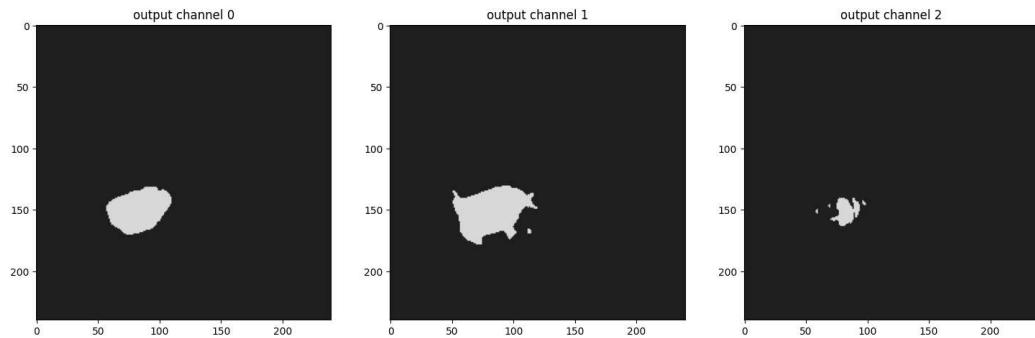
**Results achieved at 47 epochs**

---

### **MEAN ENSEMBLE**



**Fig 8.7 Ground Truth**



**Fig 8.8 Model Output**

#### **Dice Scores:**

Metric on original image spacing: **0.7645719647407532**

**metric\_tc: 0.8091**

**metric\_wt: 0.9032**

**metric\_et: 0.5814**

## **8.2 RESULT FOR CLASSIFICATION MODELS**

Ensembling is a technique in machine learning where multiple models are combined to make predictions. The idea behind ensembling is that different models may have different strengths and weaknesses, and by combining their predictions, we can achieve better overall accuracy.

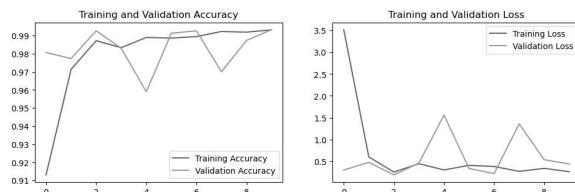
Here's a general explanation of how ensembling can improve accuracy:

- Diversity of models: The four models you used in the ensemble are likely to be diverse in terms of their architecture, algorithms, or the features they focus on. This diversity helps to capture different aspects of the data and can lead to better generalization and improved accuracy.
- Reduction of individual model errors: Each individual model may make errors or have limitations in certain situations. However, by combining their predictions, you can reduce the impact of these errors. For example, if one model performs poorly on certain samples, other models may be able to compensate and provide more accurate predictions for those cases.
- Wisdom of the crowd: Ensembling leverages the "wisdom of the crowd" principle. By combining predictions from multiple models, you can effectively average out biases or mistakes made by individual models. This can lead to more robust and accurate predictions overall.
- Overfitting reduction: Ensembling can help reduce the risk of overfitting, which occurs when a model becomes too specialized to the training data and fails to generalize well to unseen data. By combining multiple models, you introduce more variability and prevent overfitting to some extent.
- Improved stability: Ensembling can enhance the stability of predictions. If one model's predictions are highly sensitive to small changes in the input data, the ensemble can mitigate this by incorporating predictions from other models, resulting in a more reliable and stable prediction.

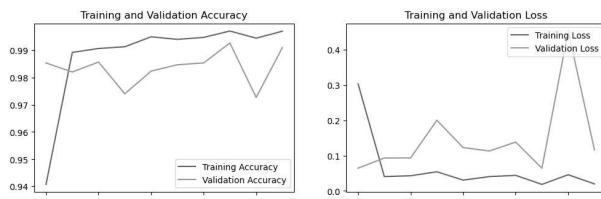
## 1. Acute Lymphoblastic Leukemia

**Table 8.1 Result for Acute Lymphoblastic Leukemia using DenseNet201**

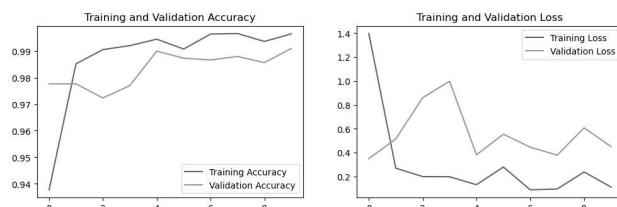
Models	Recall	Precision	F1Score
DenseNet201	<b>0.9903333333333333</b>	<b>0.9904421811723344</b>	<b>0.9903241375685572</b>
MobileNetV3 Small	<b>0.9916666666666667</b>	<b>0.991678857124651</b>	<b>0.9916619096443816</b>
VGG19	<b>0.9913333333333333</b>	<b>0.9913810676610213</b>	<b>0.99133643551295</b>
ResNet50V2	<b>0.7963333333333333</b>	<b>0.8631494378615919</b>	<b>0.7981504464349779</b>
Ensemble	<b>0.998</b>	<b>0.998006178550939</b>	<b>0.9979997690422237</b>



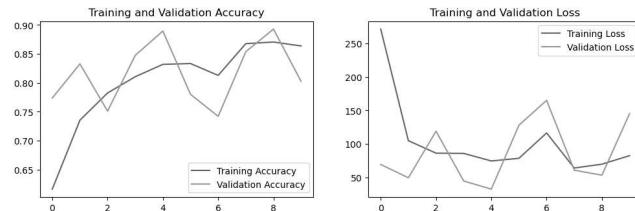
**Fig 8.9 Result for Acute Lymphoblastic Leukemia using DenseNet201**



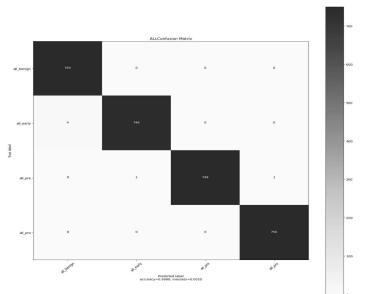
**Fig 8.10 Result for Acute Lymphoblastic Leukemia using MobileNetV3Small**



**Fig 8.11 Result for Acute Lymphoblastic Leukemia using VGG19**



**Fig 8.12 Result for Acute Lymphoblastic Leukemia using ResNet50V2**

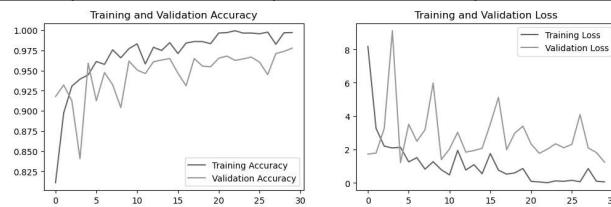


**Fig 8.13 Result for Acute Lymphoblastic Leukemia using Ensemble**

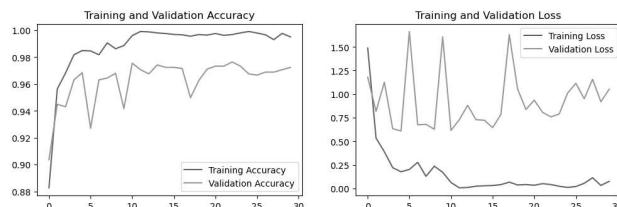
## 2. Brain Cancer

**Table 8.2 Result for Brain Cancer using DenseNet201**

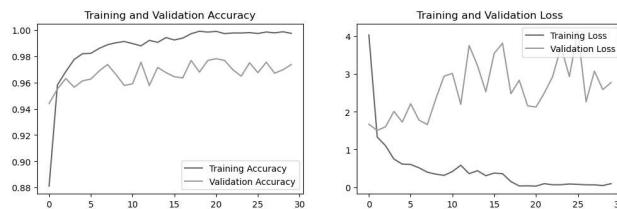
Models	Accuracy	Recall	Precision	F1Score
DenseNet201	0.95955555555556	0.95955555555556	0.959682605299268	0.9595922637062334
MobileNetV3Small	0.9697777777777777	0.9697777777777777	0.9699118252039262	0.9697824194666587
VGG19	0.9684444444444444	0.9684444444444444	0.9688725780364567	0.9683784986919409
ResNet50V2	0.8742222222222222	0.8742222222222222	0.8893087706148176	0.8757004946044663
Ensemble	0.9808888888888889	0.9808888888888889	0.980974081159352	0.9808954375713036



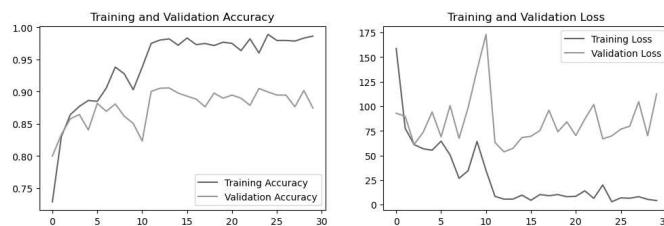
**Fig 8.14 Result for Brain Cancer using DenseNet201**



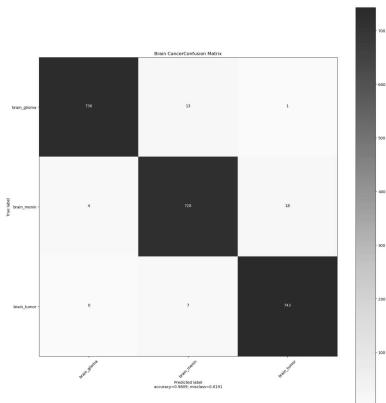
**Fig 8.15 Result for Brain Cancer using MobileNetv3Small**



**Fig 8.16 Result for Brain Cancer using VGG19**



**Fig 8.17 Result for Brain Cancer using ResNet50V2**



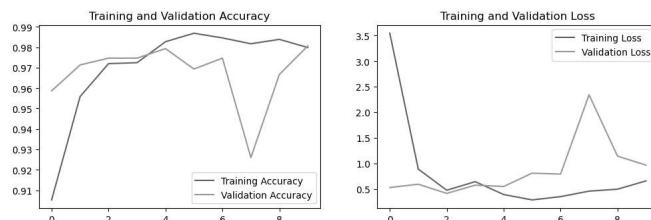
**Fig 8.18 Result for Brain Cancer using Ensemble**

### 3. Breast Cancer

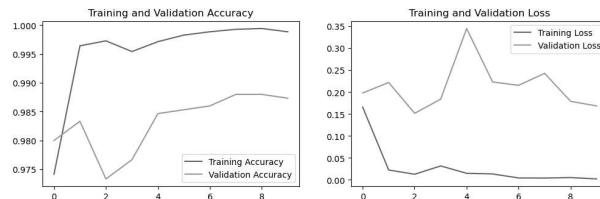
**Table 8.3 Result for Breast Cancer using DenseNet201**

Models	Accuracy	Recall	Precision	F1Score
DenseNet201	0.98	0.98	0.9806699470907005	0.9799930286819941

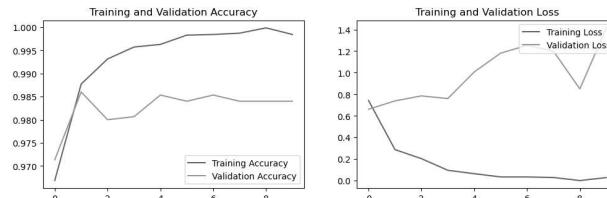
MobileNetV3Small	0.9926666666 666667	0.9926666666 666667	0.9926675425 200756	0.9926666634 07406
VGG19	0.9913333333 333333	0.9913333333 333333	0.9914040955 230887	0.9913330213 22101
ResNet50V2	0.7926666666 666666	0.7926666666 666666	0.8479715662 049592	0.7840876534 70172
Ensemble	0.9973333333 333333	0.9973333333 333333	0.9973474801 061007	0.997333143 702354



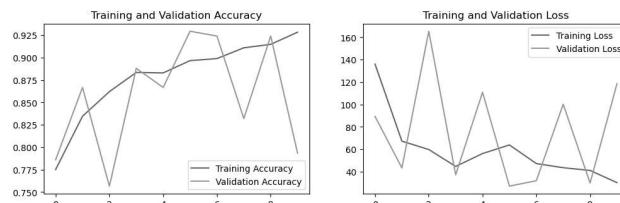
**Fig 8.19 Result for Breast Cancer using DenseNet201**



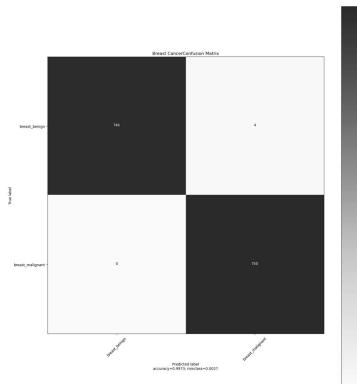
**Fig 8.20 Result for Breast Cancer using MobileNetv3Small**



**Fig 8.21 Result for Breast Cancer using VGG19**



**Fig 8.22 Result for Breast Cancer using ResNet50V2**

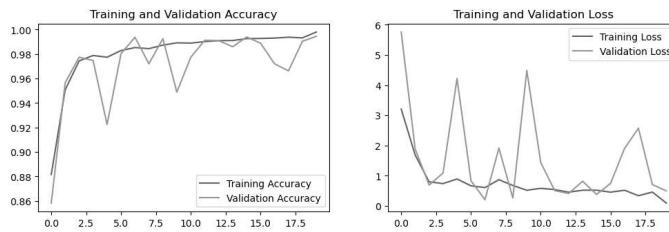


**Fig 8.23 Result for Breast Cancer using Ensemble**

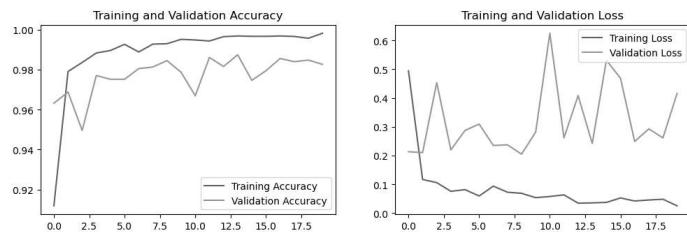
#### 4. Cervical Cancer

**Table 8.4 Result for Cervical Cancer using DenseNet201**

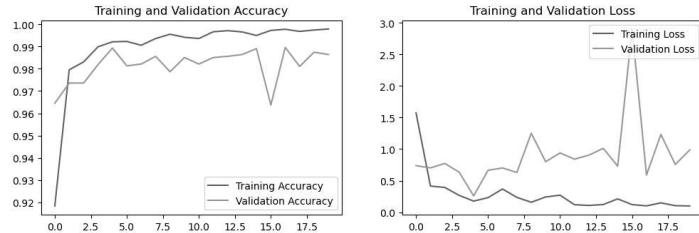
Models	Accuracy	Recall	Precision	F1Score
DenseNet201	0.9944	0.9944	0.99446302996 9946	0.99440384072 41459
MobileNetV3Small	0.98266666666 66667	0.98266666666 66667	0.98293799487 01118	0.98262953557 47524
VGG19	0.98906666666 66666	0.98906666666 66666	0.98916943644 07169	0.98907777524 61215
ResNet50V2	0.8512	0.8512	0.88191196514 5657	0.84764221667 9388
Ensemble	0.99813333333 33333	0.99813333333 33333	0.99813581607 50206	0.99813333072 40217



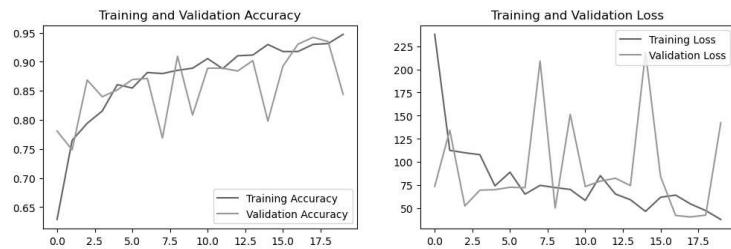
**Fig 8.24 Result for Cervical Cancer using DenseNet201**



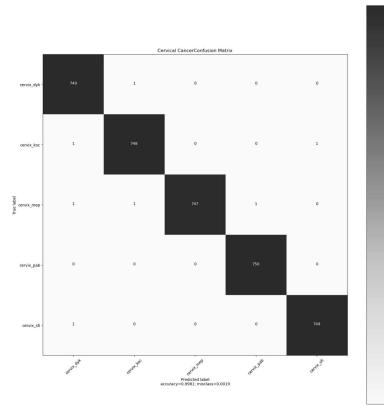
**Fig 8.25 Result for Cervical Cancer using MobileNetV3Small**



**Fig 8.26 Result for Cervical Cancer using VGG19**



**Fig 8.27 Result for Cervical Cancer using ResNet50V2**

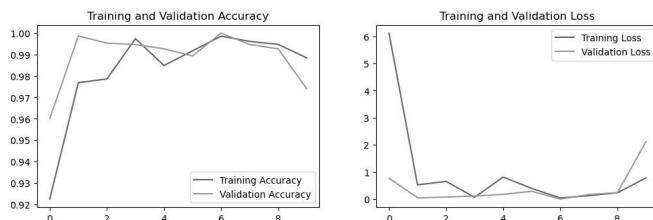


**Fig 8.28 Result for Cervical Cancer using Ensemble**

## 5. Kidney Cancer

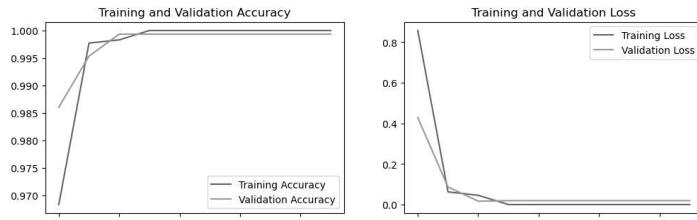
**Table 8.5 Result for Kidney Cancer using DenseNet201**

Models	Recall	Precision	F1Score
DenseNet201	0.9426666666666667	0.997179125528914	0.9691569568197395
MobileNetV3Small	1.0	1.0	1.0
VGG19	1.0	1.0	1.0
ResNet50V2	0.9933333333333333	1.0	0.9966555183946488
Ensemble	1.0	1.0	1.0

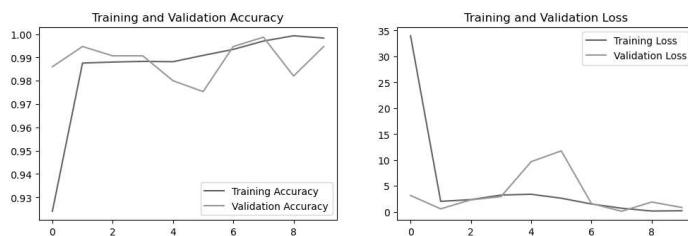


**Fig 8.29 Result for Kidney Cancer using DenseNet201**

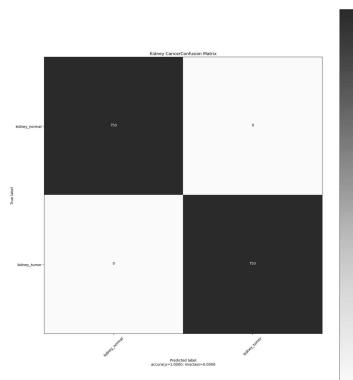
**Fig 8.30 Result for Kidney Cancer using MobileNetv3Small**



**Fig 8.31 Result for Kidney Cancer using VGG19**



**Fig 8.32 Result for Kidney Cancer using ResNet50V2**

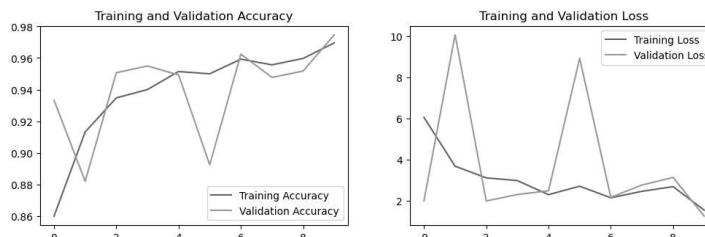


**Fig 8.33 Result for Kidney Cancer using Ensemble**

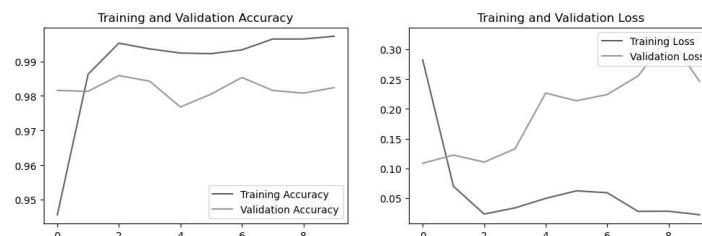
## 6. Lung and Colon Cancer

**Table 8.6 Result for Lung & Colon Cancer using DenseNet201**

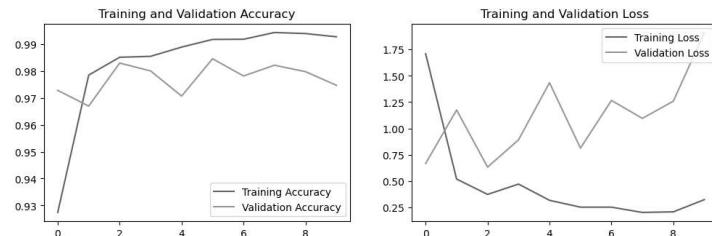
Models	Recall	Precision	F1Score
DenseNet201	0.9728	0.9729875598804594	0.972848129781705
MobileNetV3Small	0.9864	0.9864941910877201	0.9864197297322221
VGG19	0.9733333333333334	0.974243236543145	0.9732906860342894
ResNet50V2	0.868	0.8908159830262765	0.8611486751050
Ensemble	0.9930666666666667	0.9930835256960889	0.9930694579629511



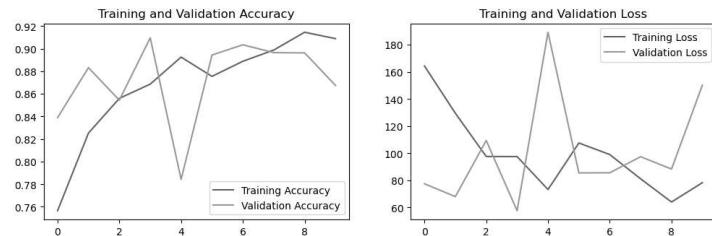
**Fig 8.34 Result for Lung & Colon Cancer using DenseNet201**



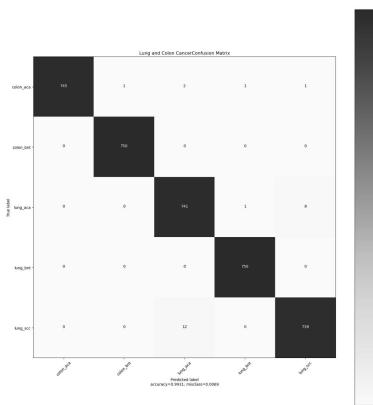
**Fig 8.35 Result for Lung & Colon Cancer using MobileNetv3Small**



**Fig 8.36 Result for Lung & Colon Cancer using VGG19**



**Fig 8.37 Result for Lung & Colon Cancer using ResNet50V2**



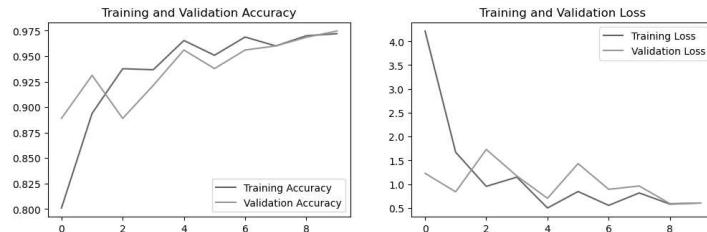
**Fig 8.38 Result for Lung & Colon Cancer using Ensemble**

## 7. Lymphoma

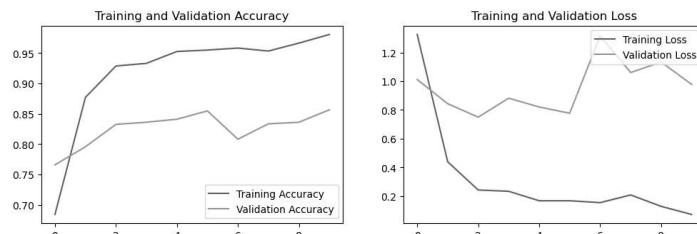
**Table 8.7 Result for Lymphoma using DenseNet201**

Models	Accuracy	Recall	Precision	F1Score
DenseNet201	0.9688888888 888889	0.9688888888 888889	0.9689020280 935331	0.9688874465 479973
MobileNetV3Small	0.8613333333 333333	0.8613333333 333333	0.8637497120 01629	0.8618748161 763887
VGG19	0.8417777777 777777	0.8417777777 777777	0.8426514454 459834	0.8415871408 756553
ResNet50V2	0.6862222222	0.6862222222	0.7781091819	0.6394392144

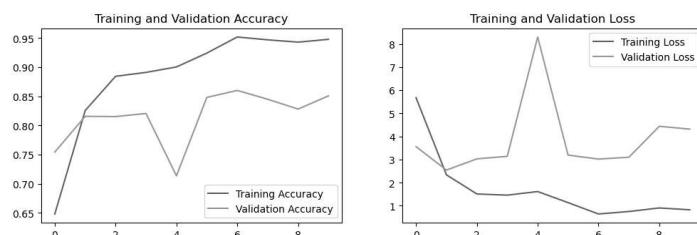
	222222	222222	31267	701045
Ensemble	0.9702222222 222222	0.9702222222 222222	0.9702118355 619148	0.9701907890 979771



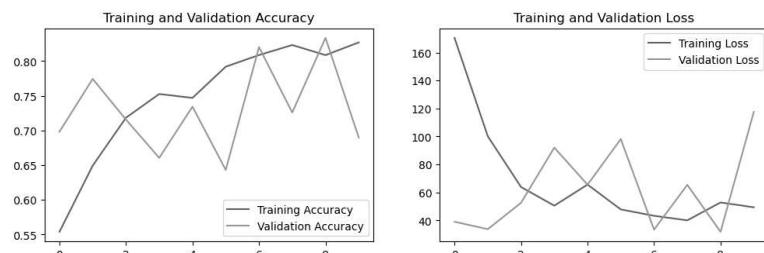
**Fig 8.39 Result for Lymphoma using DenseNet201**



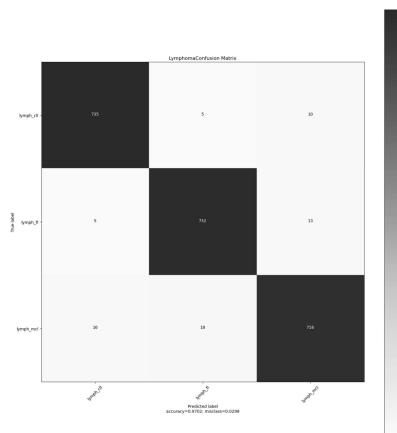
**Fig 8.40 Result for Lymphoma using MobileNetV3Small**



**Fig 8.41 Result for Lymphoma using VGG19**



**Fig 8.42 Result for Lymphoma using ResNet50V2**

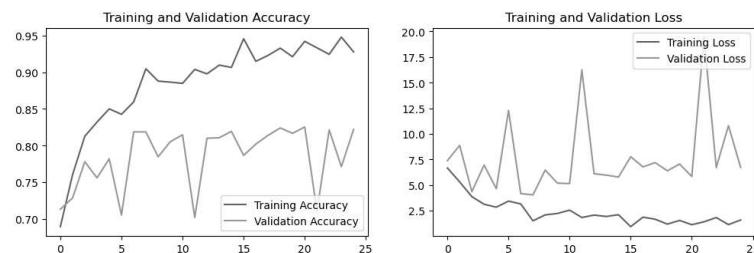


**Fig 8.43 Result for Lymphoma using Ensemble**

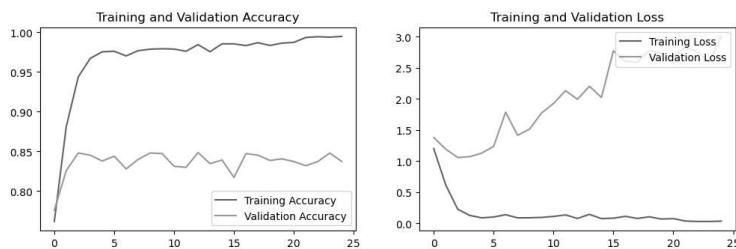
## 8. Oral Cancer

**Table 8.8 Result for Oral Cancer using DenseNet201**

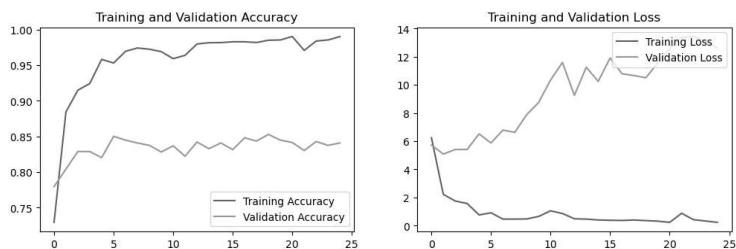
Models	Recall	Precision	F1Score
DenseNet201	0.8362183754993342	0.8406859442513094	0.835679673824873
MobileNetV3Small	0.8495339547270306	0.8495364337044347	0.8495336879432624
VGG19	0.8302263648468708	0.8302269503546099	0.8302262895927
ResNet50V2	0.8055925432756325	0.8071807301923951	0.8053409365280007
Ensemble	0.9001331557922769	0.9006901425205559	0.9000984382898342



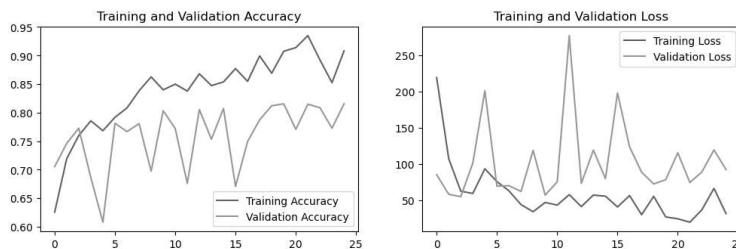
**Fig 8.44 Result for Oral Cancer using DenseNet201**



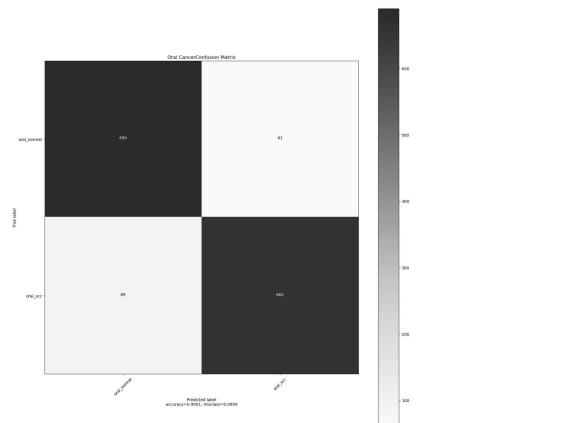
**Fig 8.45 Result for Oral Cancer using MobileNetV3Small**



**Fig 8.46 Result for Oral Cancer using VGG19**



**Fig 8.47 Result for Oral Cancer using ResNet50V2**



**Fig 8.48 Result for Oral Cancer using Ensemble**

# **CHAPTER 9**

## **CONCLUSION**

We have managed to build 2 models and tried to ensemble it to segment brain tumors and achieved desirable results. Further improvements can be done using a bigger dataset with more computational resources. Cross Validation ensemble could achieve even better results. Swin-UNETR architecture can be tried out with a better version of UNETR. Better validation data can be obtained with the help of famous radiologists in the world. We have just used models to ensemble, similarly ensemble of multiple models could lead to better dice scores.

In summary, ensemble-based approaches offer significant advantages in medical image classification and segmentation. They enhance the performance, robustness, and generalization capabilities of the system, enabling more accurate diagnoses and treatment decisions. Leveraging the power of ensembles can lead to improved healthcare outcomes and contribute to advancements in medical imaging technology.

To conclude, the detection of cancer using machine learning techniques offers several advantages that can significantly impact healthcare and improve patient outcomes. Traditional cancer diagnosis often requires extensive manual analysis and interpretation of medical images and biopsy samples, which can be time-consuming. Machine learning algorithms can automate and expedite the diagnostic process, analyzing images and data in a fraction of the time, thereby reducing waiting times for patients and enabling faster treatment initiation.

For future scope, we will focus on using more and potentially better models and create a bigger ensemble for medical image classification.

## REFERENCES

- [1] Abadi, Martín et al., “Tensorflow: Large-scale machine learning on heterogeneous distributed systems” (2016).
- [2] Abraham, Nabila and Naimul Mefraz Khan, “A novel focal tversky loss function with improved attention u-net for lesion segmentation” (2019).
- [3] Anithadevi, D and K Perumal, “A hybrid approach based segmentation technique for brain tumor in MRI Images” (2016).
- [4] Hatamizadeh A, Tang Y, “Unetr: Transformers for 3d medical image segmentation” (pp. 574-584), (2022).
- [5] Myronenko A, “3D MRI brain tumor segmentation using autoencoder regularization” (2019).
- [6] Gu, Shanqing, Manisha Pednekar and Robert Slater, “Improve Image Classification Using Data Augmentation and Neural Networks” (2019).
- [7] Salçin, Kerem et al, “Detection and classification of brain tumors from MRI images using faster R-CNN” (2019).
- [8] Jonas Prellberg, Oliver Kramer, “Acute Lymphoblastic Leukemia Detection from Microscopic Images Using Weighted Ensemble of Convolutional Neural Networks” (2019).
- [9] Marco Aurélio Granero, Cristhian Xavier Hernández, Marcos Eduardo Valle, “Quaternion-Valued Convolutional Neural Network Applied for Acute Lymphoblastic Leukemia Diagnosis” (2021).
- [10] Md Zahangir Alom, Chris Yakopcic, Tarek M. Taha, Vijayan K. Asari, “Breast Cancer Classification from Histopathological Images with Inception Recurrent Residual Convolutional Neural Network” (2018).

- [11] Aditya Golatkar, Deepak Anand, Amit Sethi, “Classification of Breast Cancer Histology using Deep Learning” (2017).
- [12] Rishav Pramanik, Momojit Biswas, et al, “A fuzzy distance-based ensemble of deep models for cervical cancer detection” (2019).
- [13] Nanditha B R, Geetha Kiran A Sanathkumar M P, “Oral Cancer Detection using Machine Learning and Deep Learning Techniques” (2021).
- [14] Panwar, S.A., Munot, M.V., Gawande, S. and Deshpande, “A reliable and an efficient approach for diagnosis of brain tumor using transfer learning” (pp.283-294), (2021).
- [15] Ullah, N., Khan, J.A., Khan, M.S., Khan, W., Hassan, I., Obayya, M., Negm, N. and Salama, A.S, “An Effective Approach to Detect and Identify Brain Tumors Using Transfer Learning. Applied Sciences” (2022).
- [16] Dalia Alzu’bi,Malak Abdullah, et.al, “Kidney Tumor Detection and Classification Based on Deep Learning Approaches: A New Dataset in CT Scans” (2022).
- [17] Md. Alamin Talukder, Md. Manowarul Islam, et.al, “Machine Learning-based Lung and Colon Cancer Detection using Deep Feature Extraction and Ensemble Learning” (2022).
- [18] Satvik Garg, Somya Garg, “Prediction of lung and colon cancer through analysis of histopathological images by utilizing Pre-trained CNN models with visualization of class activation and saliency maps” (2021).
- [19] Maria Frucci, Daniel Riccio, et.al, “A Deep Learning Approach for Breast Invasive Ductal Carcinoma Detection and Lymphoma Multi-Classification in Histological Images” (2021).
- [20] Zeyad Ghaleb Al-Mekhlafi, Ebrahim Mohammed Senan, et.al, “Diagnosis of Histopathological Images to Distinguish Types of Malignant Lymphomas Using Hybrid Techniques Based on Fusion Features” (2022).