Department of Computer Science

MSc Artificial Intelligence

Academic Year 2023-2024

Segmentation of Brain MRI Using Advanced Deep Learning Models: PSPNet and UNet-EfficientNetB7

Chaudhuri Md Tausif - 2355912

A report submitted in partial fulfilment of the requirement for the degree of Master of Science

Brunel University
Department of Computer Science
Uxbridge, Middlesex UB8 3PH
United Kingdom
Tel: +44 (0) 1895 203397
Fax: +44 (0) 1895 251686

ABSTRACT

The brain is the most crucial organ in the human body, as it regulates and oversees its fundamental activities. A group of cancerous cells that have grown inside the brain is called a brain tumor. As yet, no primary factor has been shown to contribute to the development of brain tumors. Still, it is one of the most dangerous cancers in the whole world. In clinical cancer treatment, medical imaging is crucial for diagnosis, treatment selection, and monitoring therapeutic response. Magnetic Resonance Imaging (MRI), which is frequently employed in the examination of brain tumors using acquisition procedures for example the conventional and advanced, is one of the most well-known acquisition modalities. Tumor segmentation is a challenging task because the tumors vary widely in shape, size, location, and appearance among the patients. Though there are many distinct kinds of brain tumors, meningioma, pituitary, and glioma tumors, our study is basically on glioma tumors. The Glioma Tumors are two types HGG (High Grade Glioma) and LGG (Low Grade Glioma). In this research work the dataset that has been used contains only the low grade glioma tumor images. The study work compares various segmentation algorithms and their outcomes in terms of accuracy, IOU score, and dice score. For this approach, PSPNet (Pyramid Scene Parsing Network) from scratch, PSPNet with a pre-trained model and UNet with EfficientNetB-7 pre-trained model have been deployed due to their remarkable abilities to detect anomalies in images and their outstanding performance in tasks like feature extraction, Brain tumor classification, and segmentation. In terms of brain tumor segmentation, the Pre-trained models like PSPNet and UNet with EfficientNetB7 encoder performed well compared to the PSPNet model built from scratch.

The PSPNet built from scratch obtained a dice score of 0.8054, an IOU score of 0.7815, and accuracy of 99.51% on the Test dataset, dice score of 0.8093, IOU score of 0.7812 and accuracy of 99.43 % on the Train dataset and dice score of 0.8000, IOU score of 0.7752 and accuracy of 99.43% on the Validation dataset.

The PSPNet pre-trained model obtained a dice score of 0.9064, an IOU score of 0.8763, and an accuracy of 99.78% on the Test dataset, dice score of 0.9558, IOU score of 0.9294 and accuracy of 99.86% on Train dataset and dice score of 0.9350, IOU score of 0.9058 and accuracy of 99.79% on Validation dataset.

The UNet EfficientNetB7 pre-trained model obtained a dice score of 0.9262, an IOU score of 0.9007, and an accuracy of 99.83% on the Test dataset, dice score of 0.9671, an IOU score of 0.9453 and accuracy of 99.89% on Train dataset and dice score of 0.9268, IOU score of 0.8999 and accuracy of 99.82% on the Validation dataset.

ACKNOWLEDGEMENTS

I would like to express my gratitude to my mentor and supervisor DR Tahmina Zebin from the bottom of my heart, for her constant encouragement, wonderful guidance, and support during my dissertation process. Her expertise in computer science has helped to shape the direction of my study and improve the quality of this work. I am also grateful to Computer Science Department of Brunel University London for offering a conducive academic environment and tools that enabled me to successfully complete this dissertation. Finally, I'd like to thank the module leader, Dr. Sarath Dantu, for his contributions and leadership in ensuring the project's success.

I certify that the work presented in the dissertation is my own unless referenced

Signature Chaudhuri Md Tausif

Date 11/09/2024

TOTAL NUMBER OF WORDS: 13280

Table of Contents

Chapter 1: Introduction 1
1.1 Research Problem Definition 1
1.2 Research Aim and Objectives 2
1.3 Research Approach 2
1.4 Dissertation Outline 5
Chapter 2: Literature Review 6
2.1 Background Work 6
2.1.1 CNN-Based Literature 6
2.1.2 Other Deep Learning-Based Literature 7
2.1.3 Data Augmentation based Literature 9
2.1.4 Transfer Learning Based Literature 9
2.2 Research Gap 10
2.3 Summary 10
Chapter 3 Methodology 11
3.1 Dataset Collection 11
3.2 Data Pre-processing 11
3.2.1 Loading and Handling the Missing Data 11
3.2.2 Generating File Paths for Images and Masks 12
3.2.3 Creating a DataFrame for the file path 12
3.2.4 Merging Metadata with Files 12
3.2.5 Train Test Split 12
3.3 Modelling – 13
3.3.1 PSPNet (Pyramid Scene Parsing Network) 13
3.3.2 UNet-EfficientNetB7 Transfer Learning Model 22
3.4 Performance Evaluation 23
3.4.1 F1 Score 23
3.4.2 Accuracy – 24
3.4.3 Loss Function – 24
3.4.4 Dice Score / Coefficient 25
3.4.5 IoU (Intersection Over Union) Coefficient 25
3.5 Summary – 26
Chapter 4 Result 27
4.1 Results Acquired By PSPNet From Scratch – 27
4.1.1 Brain MRI Images Along with Mask Images 27
4.1.2 The Results on Train, Test and Validation Data 28
4.1.3 Predicted Images (Tumor Areas) 30

4.1.4 Visualization On Performance Metrics –.. 32

4.2 Results Acquired by PSPNet with Transfer Learning Model and UNet EfficuientNetB7 Model– 34

4.2.1 MRI Images and Mask Images –.. 34

4.2.2 Results on Train Test and Validation Data 35

4.2.3 Predicted Images By PSPNet-Resnext50_32x4d and UNet-EfficientNetB7 40

4.2.4 Visualization of Performance Metrics on Both PSPNet-Resnext50_32x4d and UNet-EfficientNetB7 43

4.3 Summary –... 54

Chapter 5: Discussion 55

5.1 Comparative analysis on prediction Brain MRI segmentation in different models 55

5.2 Comparative Analysis on F1 score and Confusion matrix in between the transfer learning models- 55

5.3 Addressing the Research Gaps- 55

5.4 Summary - 56

Chapter 6: Conclusion 57

6.1 Summary of The Dissertation – 57

6.2 Research Contributions –.. 57

6.3 Future research and development 58

6.4 Personal Reflections 59

References.. 60

Appendix A: ETHICAL APPROVAL 67

Appendix B: Code 68

Chapter 1: Introduction

1.1 Research Problem Definition

Brain tumors are one of the most common types of cancer globally, and early detection and localization are crucial for saving lives. If malignant tumors are not discovered in the early stages, patients have a very low chance of survival, as all tumors—regardless of type—are considered high-grade due to their potential to spread to other parts of the body(Sajjad et al., 2019). Brain tumors are classified into three categories by the American Brain Tumor Association and the World Health Organization based on their location, shape, and tissue structure: gliomas, meningiomas, and pituitary tumors (Nilesh Bhaskarrao Bahadure, Arun Kumar Ray and Har Pal Thethi, 2017). The majority of gliomas are malignant tumors with increased incidence, death, and cure rates (Wang et al., 2020). Hafiz et al. (2023) defined gliomas as tumors that originate in the glial cells, which support and surround neurons in the brain. Meningiomas, on the other hand, are the most common primary brain tumors, developing in the meninges—the three outer layers of the skull. Pituitary tumors develop in the pituitary gland located at the base of the brain. Correctly identifying the type of brain tumor is crucial for accurate diagnosis and patient survival. Various medical imaging techniques, such as X-rays, CT scans, and MRI (Magnetic Resonance Imaging), are used to detect brain abnormalities. These modalities provide detailed information about the brain's internal structure, soft tissues, and cells (N. Varuna Shree and Kumar, 2018). There are two types of glioma: low-grade glioma (LGG), which progresses slowly and is easier to treat, and high-grade glioma (HGG), which is aggressive and requires prompt treatment (Louis et al., 2016). High-Grade Glioma (HGG) grows faster than Low-Grade Glioma (LGG), resulting in a shorter life expectancy for individuals with HGG compared to those with LGG (Menze et al., 2015).

Determining the type of brain tumor is crucial for accurate diagnosis and patient survival. Medical imaging modalities like CT scans, MRIs, and X-rays are used to identify brain abnormalities by providing detailed information about the brain's soft tissues, cells, and internal structure (N. Varuna Shree and Kumar, 2018).

X-ray scans, using electromagnetic waves, primarily show details of broken bones. CT scans provide internal images, particularly useful for diagnosing chest and brain disorders. MRI imaging is the most effective for examining soft and nervous tissues, using radiofrequency waves without exposing patients to ionizing radiation. It is commonly used to study the brain's internal structures and detect abnormalities (Sinkus et al., 2020).

Deep learning techniques are crucial in the medical field for identifying and classifying brain tumors, helping to determine which areas of the body are damaged or affected (Sajid, Hussain and Sarwar, 2019). Deep learning models excel in object identification, classification, pattern recognition, image analysis, and medical diagnostics. Their performance and accuracy largely depend on training with diverse data samples to effectively recognize patterns in images. (Hafiz et al., 2023). Transfer learning is a deep learning technique where a model trained on a large dataset with millions of classes is applied to a new or smaller dataset. The model utilizes the features it has already learned to improve performance on the new task (Talo et al., 2019).

The main goal of using a transfer learning approach is to address time and resource limitations. Pre-trained models, having learned from large datasets, extract low-level features (like color and edges) and higher-level features from deeper layers. Since common features across images are similar, a pre-trained model can efficiently extract these from new data samples. Additional layers can then be added to uncover hidden patterns, improving outcomes. Transfer learning's key advantages include saving time, effectively working with smaller datasets, and recognizing complex patterns in new data by leveraging previously learned features (Hafiz et al., 2023).

1.2 Research Aim and Objectives

Research Aim-

The research aims to compare deep learning models, using both transfer learning and from scratch, by evaluating loss, Dice score, IOU score, and accuracy across train, test, and validation datasets.

To accomplish this, the following objectives must be met.

I. Conduct a detailed literature review to identify suitable deep learning algorithms for the research and perform an in-depth comparative analysis of these techniques for evaluation.
II. Collect the dataset on Brain MRI and pre-process it for the segmentation task.
III. Evaluate the results obtained from different deep learning algorithms and visualize the results of the methods for better understanding.

Research questions-

To further guide the study, this research is driven by the following key questions:

1. How effective are deep learning models like PSPNet and UNet with EfficientNetB7 for segmenting brain tumors in MRI images in terms of accuracy, Dice score, and IoU?

2. Does transfer learning provide a significant advantage in brain MRI segmentation compared to models trained from scratch?

3. Which specific deep learning architecture—PSPNet or UNet EfficientNetB7—shows better generalisability and performance across training, validation, and test datasets?

1.3 Research Approach

This study employs a dual strategy, combining traditional deep learning model training and transfer learning. First, PSPNet (Pyramid Scene Parsing Network) is used, known for its high accuracy in semantic segmentation tasks (Schumacher, 2023). PSPNet's architecture is designed to capture both local and global context through its pyramid pooling module, making it highly effective in medical imaging tasks where distinguishing between abnormal and healthy tissues is often challenging (Schumacher, 2023). Training PSPNet from scratch allows us to observe how well the model learns the unique properties of brain MRI images without relying on any prior knowledge.

Deep learning models trained from scratch often require large datasets to perform optimally, which can be a disadvantage in the medical field where data is often limited (Hafiz et al., 2023). To address this, transfer learning is applied using a pre-trained version of PSPNet with the ResNeXt50-32x4d architecture, trained on the ImageNet dataset. This approach allows the model to use previously learned feature extraction capabilities for more specific tasks like brain tumor segmentation (Talo et al., 2019). This strategy aims to determine if using pre-trained models can significantly reduce training time while maintaining or even improving accuracy.

Additionally, UNet EfficientNetB7 is utilized, combining the segmentation efficiency of the UNet architecture with the powerful feature extraction capabilities of EfficientNetB7 (Tan and Le, 2019). UNet's encoder-decoder architecture captures both high- and low-level features, which is essential for accurately segmenting complex structures like brain tumors (Ronneberger, Fischer and Brox, 2015). The research compares the performance of transfer learning and training from scratch to assess their effectiveness in medical image segmentation tasks.

The Flow diagram below delineates about the research methodology –

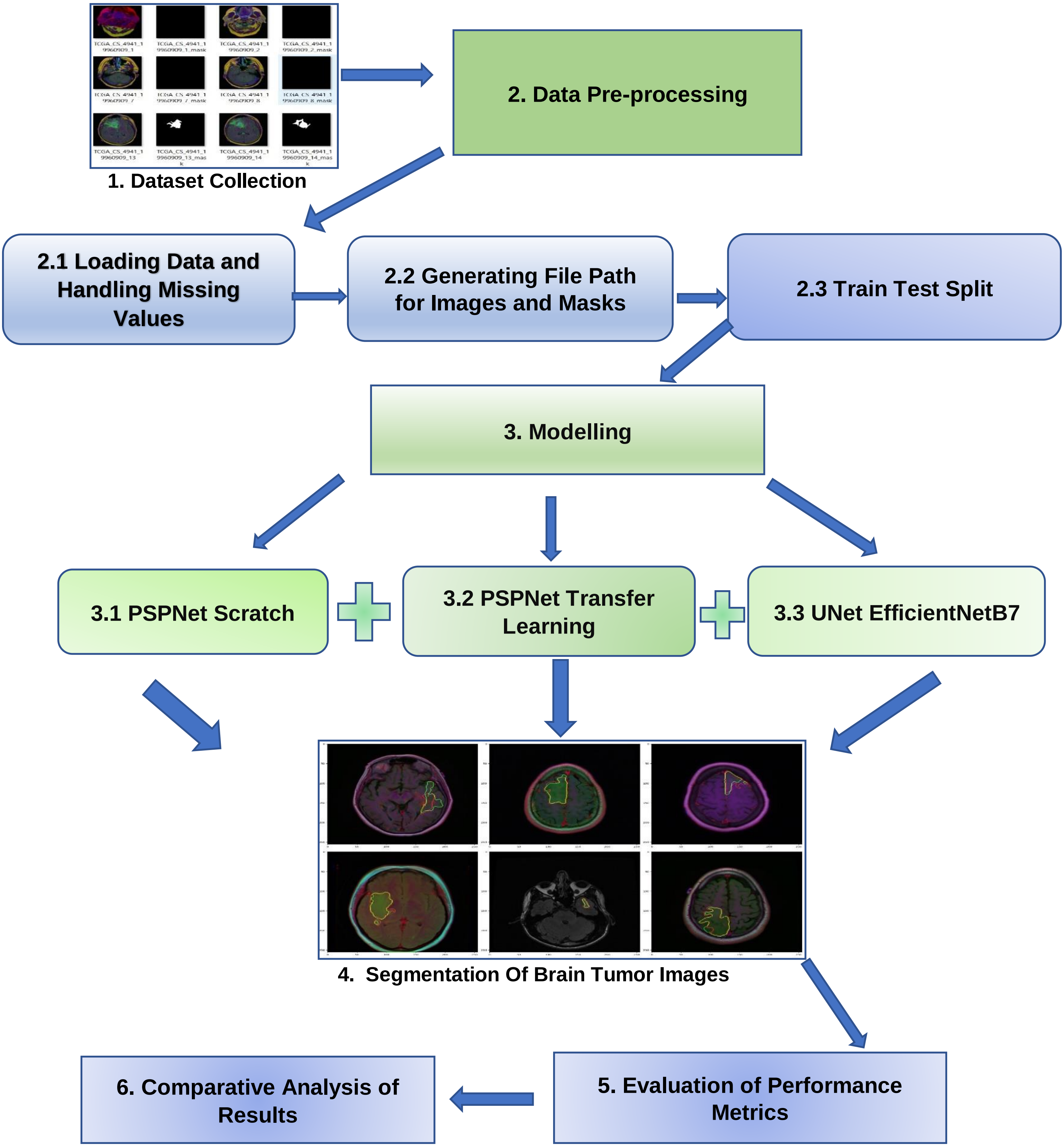

1. Dataset Collection
2. Data Pre-processing
2.1 Loading Data and Handling Missing Values
2.2 Generating File Path for Images and Masks
2.3 Train Test Split
3. Modelling
3.1 PSPNet Scratch
3.2 PSPNet Transfer Learning
3.3 UNet EfficientNetB7
4. Segmentation Of Brain Tumor Images
5. Evaluation of Performance Metrics
6. Comparative Analysis of Results

Figure 1.1: Flow Diagram of Research Methodology

1.4 Dissertation Outline

This dissertation is structured into six chapters, each addressing key components of the research in detail.

Chapter 1- This chapter introduces the research by outlining the problem, aims, objectives, and methodology. It contextualizes the brain tumor segmentation problem within medical image analysis and highlights the importance of accurate tumor detection.

Chapter 2- The literature review thoroughly examines existing research on brain MRI segmentation, emphasizing the role of deep learning models. It discusses relevant studies utilizing CNNs, transfer learning, and data augmentation, highlighting research gaps this dissertation aims to address.

Chapter 3- This chapter outlines the methodology, covering data collection, pre-processing, and model deployment. It explains the selection of deep learning models, the datasets used, and the performance metrics for evaluation. Additionally, it details the hyperparameters used for training and optimizing the models.

Chapter 4- The results chapter presents the performance of the three deep learning models PSPNet from Scratch, PSPNet Transfer Learning model, and UNet EfficientNetB7 on brain MRI segmentation. The models are compared based on accuracy, Dice score, IoU score, and other metrics across different datasets to address the research questions from Chapter 1. Visualizations of segmented images are included for qualitative comparison.

Chapter 5- This chapter discusses the significance of the results of the literature reviewed in Chapter 2. It highlights the strengths and limitations of each model and provides insights into improving deep-learning techniques for medical image segmentation. Additionally, it addresses the research gaps identified in Chapter 2, with findings from this study helping to fill those gaps.

Chapter 6- The final chapter provides conclusions drawn from the research and outlines potential directions for future work.

Chapter 2: Literature Review

This chapter analyzes the available literature, investigates methodologies to brain MRI segmentation, and identifies research gaps discovered in prior studies. It surveys and discusses automatic approaches for detecting brain cancers in MRI images, as well as previous research that used various procedures or techniques.

2.1 Background Work

According to Anu Singha and Venkateswaran (2023) in many areas of brain tumor diagnosis, machine learning techniques have been widely used in medicine. Recently, researchers have proposed various deep-learning networks for brain tumor diagnostics, either for detection, segmentation, or classification. These networks use imaging modalities like MRI, CT, PET, and a combination of them.

2.1.1 CNN-Based Literature

According to Hafiz et al. (2023) In the field of brain tumor detection, numerous research articles have been published. Most research has focused on diagnosing brain tumors using traditional machine learning and image processing algorithms, with the primary goal being to identify tumors. Some research focused on identifying the type of tumor. Before deep learning, machine learning algorithms were commonly used, but feature extraction had to be done manually, which could lead to missed tumor detection due to false features. Deep learning models, like CNNs, have become increasingly popular in image classification because they can automatically extract and learn features, improving accuracy (Hafiz et al., 2023).

Some of them are described below-

Syed Ali Yazdan et al. (2022) proposed a study classifying brain tumors into four distinct classes. A non-local means filter based on fuzzy similarity was used to remove noise from images. They evaluated a multiscale CNN model and compared it to AlexNet and ResNet on a 3264-image dataset from Kaggle. The multiscale CNN outperformed the others, achieving 91.2% accuracy and a 91% F1-score. However, despite the complex design, the model required excessive time and processing power, leading to subpar results.

Saeedi et al. (2023) suggested two deep learning models for identifying different types of brain tumors: an auto-encoder convolutional network model and a 2D convolutional neural network model. They used a dataset of 3264 images with four types of tumors from Kaggle. The 2D CNN achieved 93.44% accuracy, and the auto-encoder model achieved 90.92%. They also tested various machine learning classifiers, with k-nearest neighbor achieving 86% accuracy and multilayer perceptron the lowest at 28%. However, further evaluation metrics like recall, precision, and F1-score were not computed.

Prince Priya Malla, Sahu and Alutaibi (2023) presented a brain tumor classification system based on a pretrained VGGNet architecture, using a transfer learning-based CNN model. They incorporated global average pooling at the output layer and froze convolutional layers to prevent overfitting and vanishing gradients. The model achieved 98.93% testing accuracy using the Figshare dataset. However, a drawback is the use of a simple model with a small dataset, and the approach only supports binary classification, lacking multiclass classification.

Osman Özkaraca et al. (2023) intoduced several brain tumor classification methods using dense CNN architectures, including DenseNet, VGG16, and a basic CNN. These models, based on transfer learning, were tested on a dataset of 7021 MRI images from Kaggle. The dataset was split into 80% for training and 20% for testing. The models achieved accuracies between 94% and 97%, demonstrating the framework's effectiveness. However, the primary issue with this approach is the slow response and lengthy processing time, which needs to be addressed.

Marco Antonio Gómez-Guzmán et al. (2023) conducted research comparing a generic CNN model with six pre-trained models to classify multiclass brain tumors. The models used were ResNet50, InceptionV3, InceptionResNetV2, Xception, MobileNetV2, and EfficientNetB0. The Msoud dataset, containing 7023 MRI images from Kaggle, was used for testing. InceptionV3 performed the best with 97.12% accuracy. However, key evaluation metrics like sensitivity, specificity, and F1-score were not used to evaluate the model.

Anand et al. (2023) presented a study on brain tumor classification using a weighted average ensemble deep learning model, combining VGG19, CNN with augmentation, and CNN without augmentation. The ensemble weights were further optimized using grid search. The proposed architecture achieved 98% accuracy on a dataset of low-grade glioma images.

Rasheed et al. (2023) proposed a novel CNN method for detecting brain tumors using MRI scans. They compared its performance with pre-trained models such as InceptionV3, ResNet50, MobileNetV2, VGG16, and VGG19. Their proposed algorithm achieved a 98% F1 score and 98.04% accuracy.

To effectively detect brain tumors using MRI images, Aloraini et al. (2023) presented a hybrid model combining a vision transformer (ViT) and a CNN for brain tumor detection, applied to two datasets: Brats 2018 and Figshare. Due to the absence of a non-tumor class, they used a lightweight model. Their model achieved 96.75% accuracy on the Brats 2018 dataset and 99.10% accuracy on the Figshare dataset.

Li et al. (2019) a multi-CNN approach that combines multimodal information fusion by extending 2D-CNNs to 3D-CNNs for brain cancer detection. The model was evaluated on the BraTs 2018 dataset and achieved a dice score of 0.927, demonstrating its effectiveness in detecting tumor lesions.

Kong et al. (2022) proposed a 3D fully convolutional network that incorporates both global and local attention mechanisms for brain tumor segmentation. The global attention focuses on accurately segmenting large tumor regions, while the local attention refines the features obtained from the global attention for more precise segmentation.

Ma et al. (2019) proposed a CNN-based method for automated primary nasopharyngeal carcinoma (NPC) tumor segmentation using both CT and MR images. Their results indicated that combining CT and MR modalities can lead to higher segmentation accuracy compared to using a single modality.

2.1.2 Other Deep Learning-Based Literature

Chen, Ding, and Liu (2019) extended the DeepMedic model by introducing a multi-level version called MLDeepMedic, which combines multi-level data for more accurate tumor lesion segmentation. They also proposed the dualforce training strategy, applied to both U-Net and MLDeepMedic models. Additionally, they used a Multi-Layer Perceptron-based post-processing step to enhance the model. Experiments were conducted on the BRATS 2017 and

BRATS 2015 datasets. For BRATS 2017, the dice scores were 85.03 for necrotic/non-enhancing tumors, 77.46 for edema, and 70.77 for enhancing tumors. For BRATS 2015, the scores were 85, 70, and 63, respectively, for these categories.

Muhammad Junaid Ali, Raza, and Ahmad Raza Shahid (2021) introduced the multi-level Kronecker-based CNN (MLKCNN), which extracts information at various levels to capture both local and global context. The use of Kronecker convolution addresses the issue of missing pixels caused by dilated convolution. They also employed a post-processing strategy that combines conditional random fields (CRF) and connected component analysis (CCA), improving segmentation performance. The model achieved dice scores of 0.74 for enhancing tumors, 0.90 for whole tumors, and 0.83 for tumor core (TC).

Khaled Bousabarah et al. (2020) carried out several U-Net models, including a modified U-Net (moU-Net), a smaller lesion-trained U-Net (sU-Net), and a conventional U-Net (cU-Net). They evaluated the performance of these models using brain metastasis data from a private clinical dataset, achieving sensitivity results ranging from 0.77 to 0.82.

Hu et al. (2020) proposed a 3D residual network called "Brain SegNet" for automatic voxel-wise segmentation of brain tumor or ischemic stroke areas in 3D MRIs. The model was evaluated on the ISLES 2017 dataset for stroke lesion detection and the BRATS 2015 dataset for brain tumor segmentation. It achieved a dice score of 0.30±0.22 for stroke lesion detection and 86.0 for whole tumor detection in brain tumors.

Aswani and D. Menaka (2021) proposed a dual autoencoder network with latent space optimization and SVD for tumor detection. The final decoder uses adjusted latent features to distinguish tumor pixels from background pixels based on reconstruction error. Instead of traditional square patches, the network uses horizontal and vertical patches during training. The proposed method achieved a dice similarity score of 0.84 in the experiment.

Guan et al. (2022) proposed a 3D MRI-based brain tumor segmentation framework called AGSE-VNet. They introduced an Attention Guide Filter (AG) module in each decoder and a Squeeze and Excite (SE) module in each encoder to enhance useful information and suppress irrelevant details. In the BraTS 2020 challenge, AGSE-VNet achieved dice scores of 0.68 for the entire tumor, 0.85 for the tumor core, and 0.70 for the augmented tumor.

Zhang et al. (2021b) introduced a Multi-scale Mesh Aggregation Network (MSMANet) that uses an enhanced Inception module for extracting and aggregating features instead of the typical convolution. They also applied a mesh aggregation approach to improve shallow features and reduce the semantic gap. Experiments on BraTS 2018 MRI scans resulted in dice scores of 0.81 for the core tumor, 0.89 for the total tumor, and 0.75 for the enhanced tumor.

Sun et al. (2019) analyzed glioma tumor patients using a private dataset and proposed a technique called Potential Field Segmentation (PFS). This method utilized T1c, T2, apparent diffusion constant (ADC), and fractional anisotropy (FA) images from five glioma patients. A global-PFS was applied for rough segmentation on a user-defined region of interest (ROI) for T1c, T2, and ADC images, while a local-PFS was used to exclude white matter (WM) from FA images. The experimental study achieved a dice score of 0.88±0.04.

Wu et al. (2021) proposed a generative adversarial network (GAN) model called Symmetric Driven GAN (SDGAN), designed to learn non-linear mappings between left and right brain images. After training, SDGAN was used to segment tumors and reconstruct normal brains. The model was tested on MRI samples of HGGs and LGGs from the BRATS 2012 and 2018 datasets, achieving dice scores of 0.619 for the BRATS 2018 dataset and 0.646 for the BRATS 2012 dataset.

Zhang et al. (2021a) proposed Triple Intersecting U-Nets (TIU-Nets) for brain glioma tumor segmentation, which include a multi-class segmentation U-Net (MU-Net) and a binary-class segmentation U-Net (BU-Net). BU-Net's soft-mask features for the glioma region are reused by MU-Net for multi-category segmentation. On the BRATS 2015 dataset, the model achieved dice scores of 0.85 for the whole tumor, 0.69 for the core tumor, and 0.63 for the enhanced tumor.

Zhou, He and Jia (2020) developed a 3D fully convolutional neural network called AFPNet, which features an atrous-convolution feature pyramid. AFPNet was designed for brain tumor segmentation and addresses the issue of resolution loss in feature maps caused by frequent striding and pooling operations.

Mohammad Ashraf Ottom, Hanif Abdul Rahman and Dinov (2022) introduced Znet, a method based on skip-connection, encoder-decoder, and data amplification for segmenting low-grade glioma tumors in the brain. This approach leverages the expertise of a small number of experts. The model achieved a dice score of 0.91 and an F1 score of 0.80.

Chen et al. (2019) proposed a lightweight network for real-time dense voxel brain tumor segmentation, focusing on reducing the number of parameters by using 3D shuffle units. They also incorporated atrous convolution to expand the receptive field for better image analysis.

Zhou et al. (2021) proposed an efficient ERVNet, which significantly improved network performance. However, the large number of parameters in the network also increased the GPU workload.

Liu et al. (2022) proposed a scale-adaptive MetricUNet that effectively learns more contextual information while reducing computational requirements.

Jiang, Zhai and Kong (2021) proposed a dual-branch decoding network called DDUnet. One decoding branch handles semantic flow information, while the other focuses specifically on edge flow information.

2.1.3 Data Augmentation based Literature

By applying the noise addition and shearing techniques, Amjad Rehman Khan et al. (2021) were able to expand the dataset's size and enhance the precision of tumor segmentation and classification.

Dufumier et al. (2021) used techniques such as rotation, random cropping, noise addition, translation, and blurring to augment the dataset and improve performance in predicting age and sex categorization.

Various research simultaneously improved tumor segmentation and accuracy by using elastic deformation, rotation, and scaling (Isensee et al., 2020).

2.1.4 Transfer Learning Based Literature

Md. Alamin Talukder et al. (2023) presented a deep learning approach using transfer learning with various pre-trained models for efficient brain tumor classification, utilizing the Figshare

MRI brain tumor dataset. They also highlighted the growing popularity of transfer learning in enhancing UNET's performance and addressing specific challenges in medical image segmentation (MIS).

Using the uncertainty sampling method, Hao et al. (2021) suggested a system based on transfer learning and active learning to lower the annotation cost.

To classify brain cancers, S. Deepak and Ameer (2019) used transfer learning to extract features from brain MRI images.

Neubauer et al. (2022) presented a framework that transfers knowledge from human claustrum in adult imaging and applies it to neonatal scans.

Bairagi et al. (2023) employed transfer learning to speed up the training process and concentrated on adjusting the hyperparameters.

2.2 Research Gap

Despite tremendous advances in brain tumor segmentation using deep learning algorithms, several major challenges remain.

Segmentation Accuracy- Although CNNs have attained high accuracy, there are still issues with segmentation precision, particularly when determining tumor boundaries. Some models struggle to distinguish specific tumor subregions such as necrotic, enhancing, and edematous areas. Studies like **Hafiz et al. (2023)** has identified these segmentation challenges, especially when dealing with smaller or more ambiguous tumor regions.

Computational Efficiency- The high computational cost of deep learning models continues to hinder their practical application. Many of the proposed architectures, such as **LinkNet (Ramasamy, Singh and Yuan, 2023)** and the models proposed by **Osman Özkaraca et al. (2023)**, are computationally expensive and time-consuming due to their depth and complexity.

Evaluation Metrics- Some studies, while reporting high accuracy, lack comprehensive evaluation using essential metrics such as sensitivity, specificity, recall, precision, and f1-score. For example, **Saeedi et al. (2023)** and **Marco Antonio Gómez-Guzmán et al. (2023)** did not fully evaluate their models using these metrics, limiting the ability to assess their practical utility in medical diagnostics.

2.3 Summary

This chapter has brought attention to the research gaps, particularly concerning Brain MRI Segmentation, by comparing various research projects. To accomplish the stated goal and objectives of this research, the next chapter will explain the approach utilized to create and enhance the Brain MRI Segmentation model's performance.

Chapter 3 Methodology

This chapter explains about the methods have been used in case of Brain MRI Segmentation work. It contains four section. The first section is Dataset Collection. The second section depicts about the data pre-processing. The third section is modeling and the fourth section is performance evaluation.

3.1 Dataset Collection

The Dataset we have taken from Kaggle is based on the Low-Grade Glioma (LGG) Brain tumor segmentation.

The dataset includes manual FLAIR abnormality segmentation masks along with brain MRI images provided by The Cancer Imaging Archive (TCIA). It corresponds to 110 individuals with FLAIR sequencing and genomic cluster data from The Cancer Genome Atlas (TCGA) lower-grade glioma collection. The data.csv file contains patient information and tumor genomic groupings (Buda, 2019).

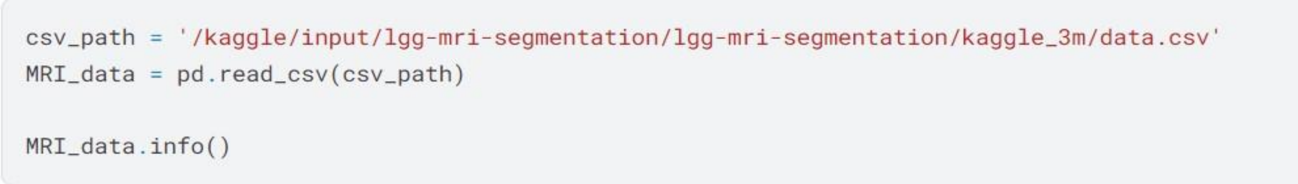
Here I have provided the Kaggle dataset link.

Kaggle - https://www.kaggle.com/datasets/mateuszbuda/lgg-mri-segmentation

3.2 Data Pre-processing

3.2.1 Loading and Handling the Missing Data

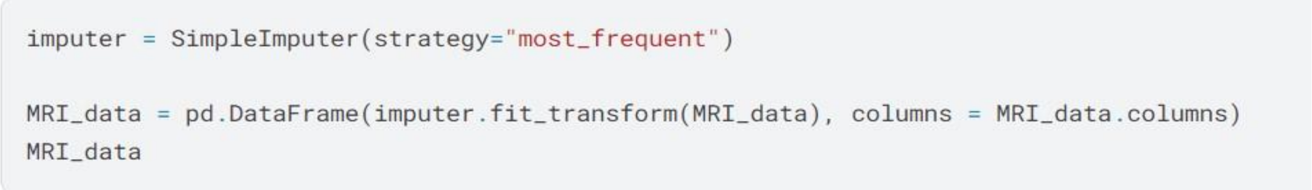
3.2.1.1 Loading the CSV Data



```
csv_path = '/kaggle/input/lgg-mri-segmentation/lgg-mri-segmentation/kaggle_3m/data.csv'
MRI_data = pd.read_csv(csv_path)

MRI_data.info()
```

The primary goal of loading the CSV data is to create a file containing MRI image metadata, such as patient IDs, scan dates, and other important details. The Python pandas library's read_csv method is used for this task.

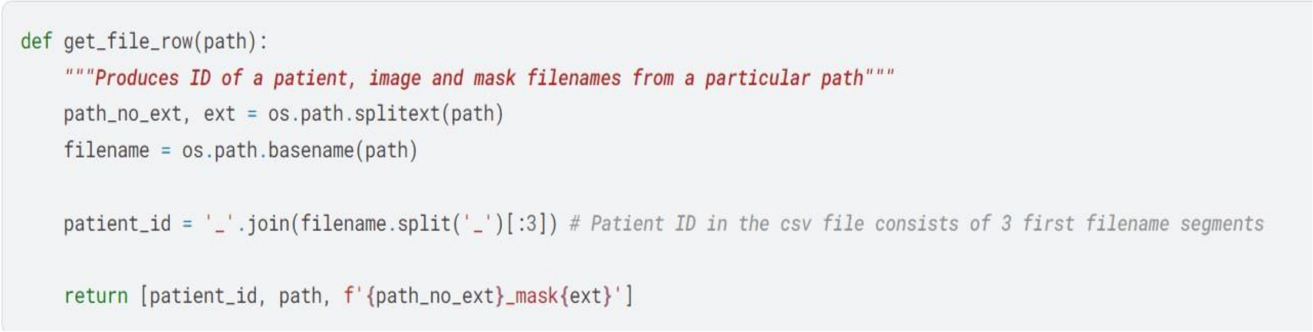
3.2.1.2 Handling The Missing Data



```
imputer = SimpleImputer(strategy="most_frequent")

MRI_data = pd.DataFrame(imputer.fit_transform(MRI_data), columns = MRI_data.columns)
MRI_data
```

The main objective is to handle missing data in a dataset. SimpleImputer, a Scikit-Learn tool, is used to fill in missing values by replacing them with the "most frequent" value from each column. This ensures no data points are missed during model training.

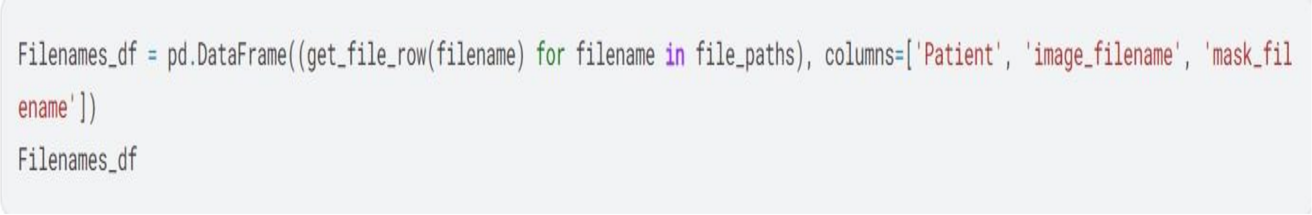
3.2.2 Generating File Paths for Images and Masks



```python
def get_file_row(path):
    """Produces ID of a patient, image and mask filenames from a particular path"""
    path_no_ext, ext = os.path.splitext(path)
    filename = os.path.basename(path)

    patient_id = '_'.join(filename.split('_')[:3]) # Patient ID in the csv file consists of 3 first filename segments

    return [patient_id, path, f'{path_no_ext}_mask{ext}']
```

This function get_file_rows() generates the corresponding mask file path and extracts the patient ID from the filename, where the patient ID is derived from the first three characters, assumed to be unique for each patient. The output is a list containing the image file path, the matching mask file path, and the patient ID.

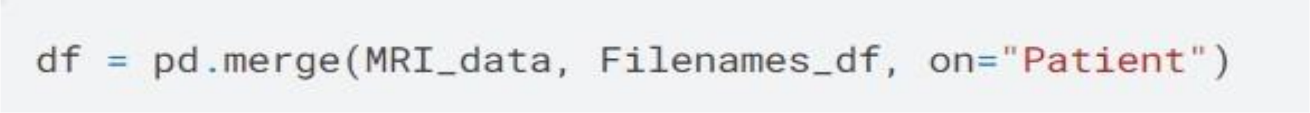
3.2.3 Creating a DataFrame for the file path



```python
Filenames_df = pd.DataFrame((get_file_row(filename) for filename in file_paths), columns=['Patient', 'image_filename', 'mask_fil
ename'])
Filenames_df
```

The goal is to create a new DataFrame called `Filenames_df` and populate it with each image's mask file path, image file path, and patient ID.

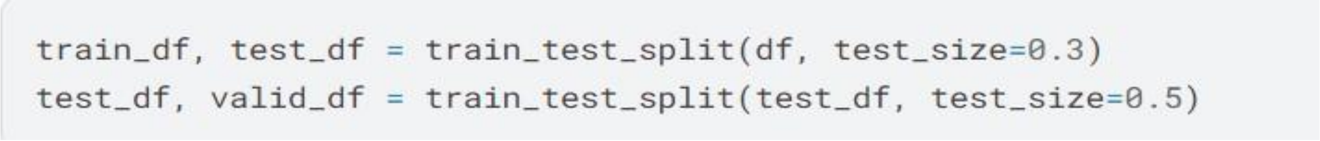
3.2.4 Merging Metadata with Files



```python
df = pd.merge(MRI_data, Filenames_df, on="Patient")
```

The objective is to merge the “Filenames_df” DataFrame with the original metadata DataFrame on the Patient column. This ensures that all pertinent information, including metadata, image, and mask file locations, is consolidated in one place for easy access during dataset construction.

3.2.5 Train Test Split



```python
train_df, test_df = train_test_split(df, test_size=0.3)
test_df, valid_df = train_test_split(test_df, test_size=0.5)
```

The main goal is to split the dataset into Train, Test, and Validation sets. First, 70% of the data is used for training and 30% for testing. Then, the test data is further divided, with 50% allocated to validation and 50% to the final test set.

3.3 Modelling –

For this research, we have deployed three advanced deep-learning models for Brain MRI segmentation: PSPNet (Pyramid Scene Parsing Network) from scratch, PSPNet- ResNeXt50-32x4d and UNet EfficientNetB7.

PSPNet, with its pyramid pooling module, captures both global and local contextual information, making it ideal for tasks that require a thorough understanding of spatial context. (Schumacher, 2023).

UNet EfficientNetB7 combines EfficientNetB7's strong feature extraction with the UNet architecture, known for its efficiency in biomedical image segmentation. Both models are trained on the same dataset with hyperparameters customized for each architecture (Hafiz et al., 2023).

3.3.1 PSPNet (Pyramid Scene Parsing Network)

For PSPNet, we have deployed this model in two different ways. The first one is PSPNet from scratch and the second is PSPNet using transfer learning technique.

PSPNet (Pyramid Scene Parsing Network) is a powerful semantic segmentation model that gathers global context information through a pyramid parsing module using different-region-based context aggregation. By combining local and global clues, it aims to improve the reliability of final predictions (Schumacher, 2023).

3.3.1.1 PSPNet from Scratch

How PSPNet works –

PSPNet uses a pre-trained Convolutional Neural Network (CNN) with the dilated network method to extract feature maps from an input image, reducing the final feature map to one-eighth of the original image size. The pyramid pooling module is then applied to gather context data on top of this feature map (Schumacher, 2023).

The pyramid pooling module in PSPNet captures diverse object scales by combining feature maps with different receptive fields and resolutions. These multiple scales form a "pyramid," which gives the network its name (Schumacher, 2023).

The final prediction map is created by concatenating the feature map from the pooling module with the original feature map and passing it through a convolution layer. This output semantically segments the input image, assigning each pixel a label corresponding to a class from the training data (Schumacher, 2023).

A. Encoder Decoder Architecture for Semantic Segmentation

Most semantic segmentation models have two main components: an encoder and a decoder. The encoder extracts feature from the image, while the decoder makes final predictions about the class of each pixel. This encoder-decoder architecture is commonly used in segmentation tasks (Developer, 2016).

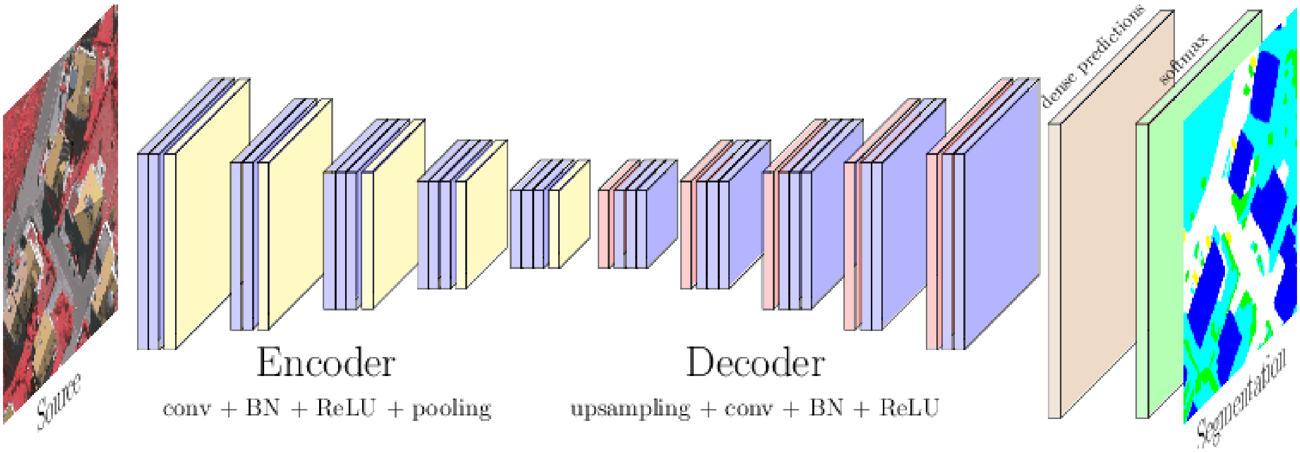


Figure 3.1: Encoder Decoder Segmentation

B. PSPNet Encoder

The CNN backbone with dilated convolutions and the pyramid pooling module are both included in the PSPNet encoder (Developer, 2016).

C. Dilated Convolution

To increase the receptive field, dilated convolution layers are used instead of standard convolution layers in the last layers of the backbone. The last two backbone blocks contain these dilated layers, resulting in more detailed features in the final feature map (Developer, 2016).

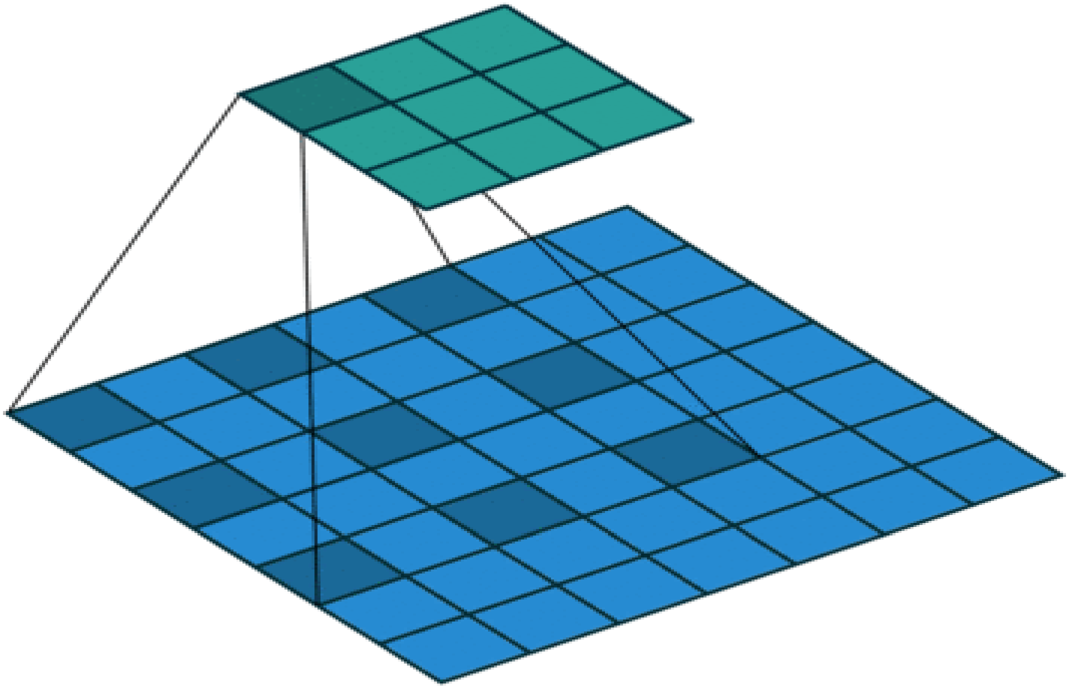


Figure 3.2: Dilated Convolution

D. Pyramid Pooling Layer

The pyramid pooling module is the key component of this model, helping capture the global context of the image. It allows the model to classify pixels based on the global information present in the image (Developer, 2016).

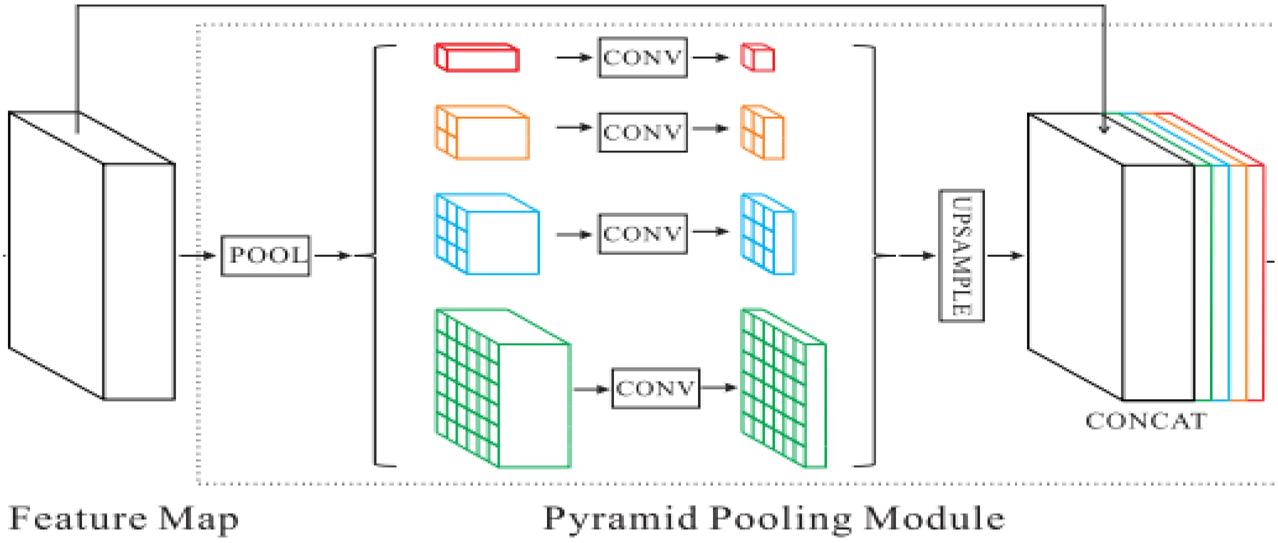


Figure 3.3: Pyramid Pooling Layer

The backbone feature map is divided into pools of varying sizes (6, 3, 2, and 1) and processed through a 1x1 convolution layer before being upsampled to match the original feature map's dimensions. These upsampled maps are then concatenated with the original feature map and sent to the decoder. This method aggregates context from different scales. The 1x1 feature represents the most coarsely pooled map with all information in a single spatial location, while larger pyramid sizes (6x6) capture higher-resolution features. Smaller pyramid sizes (1 or 2) capture larger, low-resolution features, while larger pyramid sizes (6-8) capture high-resolution details (Developer, 2016).

E. PSPNet Decoder

After the encoder extracts the image's features, the decoder processes them and generates predictions. The decoder is another network responsible for this translation. However, a decoder with 8x upsampling lacks learnable parameters, which is a limitation (Developer, 2016).

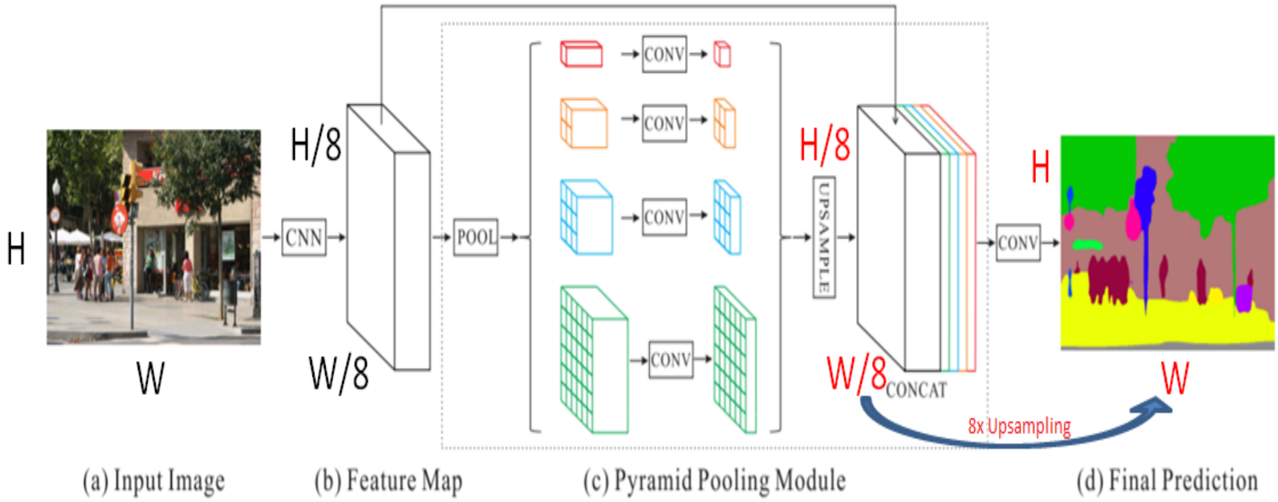
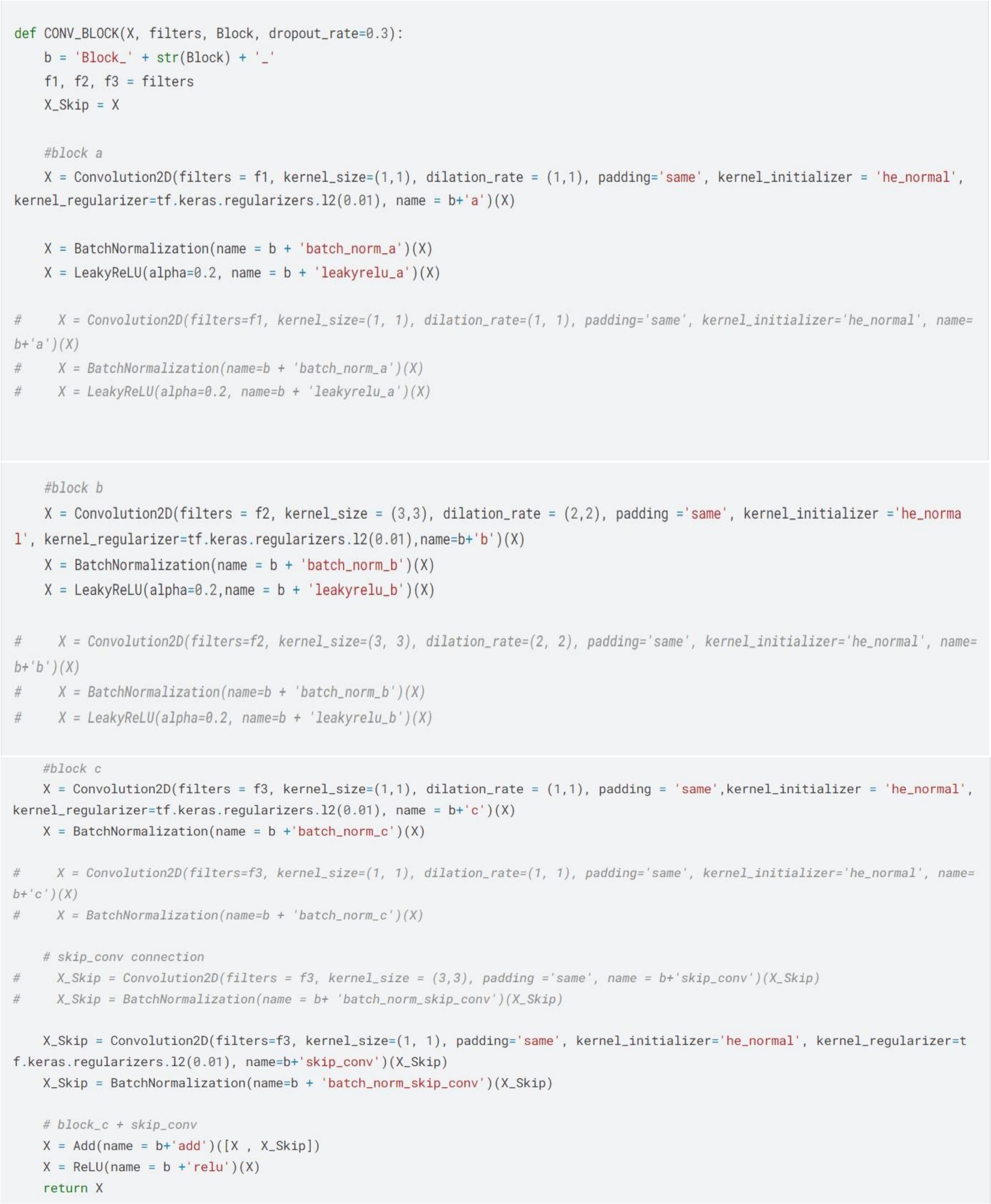


Figure 3.4: PSPNet with with 8 upsampling decoders

The code structure of PSPNet –

A. The Convolution Block



```
def CONV_BLOCK(X, filters, Block, dropout_rate=0.3):
    b = 'Block_' + str(Block) + '_'
    f1, f2, f3 = filters
    X_Skip = X

    #block a
    X = Convolution2D(filters = f1, kernel_size=(1,1), dilation_rate = (1,1), padding='same', kernel_initializer = 'he_normal',
kernel_regularizer=tf.keras.regularizers.l2(0.01), name = b+'a')(X)

    X = BatchNormalization(name = b + 'batch_norm_a')(X)
    X = LeakyReLU(alpha=0.2, name = b + 'leakyrelu_a')(X)

#     X = Convolution2D(filters=f1, kernel_size=(1, 1), dilation_rate=(1, 1), padding='same', kernel_initializer='he_normal', name=
b+'a')(X)
#     X = BatchNormalization(name=b + 'batch_norm_a')(X)
#     X = LeakyReLU(alpha=0.2, name=b + 'leakyrelu_a')(X)



    #block b
    X = Convolution2D(filters = f2, kernel_size = (3,3), dilation_rate = (2,2), padding ='same', kernel_initializer ='he_norma
l', kernel_regularizer=tf.keras.regularizers.l2(0.01),name=b+'b')(X)
    X = BatchNormalization(name = b + 'batch_norm_b')(X)
    X = LeakyReLU(alpha=0.2,name = b + 'leakyrelu_b')(X)

#     X = Convolution2D(filters=f2, kernel_size=(3, 3), dilation_rate=(2, 2), padding='same', kernel_initializer='he_normal', name=
b+'b')(X)
#     X = BatchNormalization(name=b + 'batch_norm_b')(X)
#     X = LeakyReLU(alpha=0.2, name=b + 'leakyrelu_b')(X)

    #block c
    X = Convolution2D(filters = f3, kernel_size=(1,1), dilation_rate = (1,1), padding = 'same',kernel_initializer = 'he_normal',
kernel_regularizer=tf.keras.regularizers.l2(0.01), name = b+'c')(X)
    X = BatchNormalization(name = b +'batch_norm_c')(X)

#     X = Convolution2D(filters=f3, kernel_size=(1, 1), dilation_rate=(1, 1), padding='same', kernel_initializer='he_normal', name=
b+'c')(X)
#     X = BatchNormalization(name=b + 'batch_norm_c')(X)

    # skip_conv connection
#     X_Skip = Convolution2D(filters = f3, kernel_size = (3,3), padding ='same', name = b+'skip_conv')(X_Skip)
#     X_Skip = BatchNormalization(name = b+ 'batch_norm_skip_conv')(X_Skip)

    X_Skip = Convolution2D(filters=f3, kernel_size=(1, 1), padding='same', kernel_initializer='he_normal', kernel_regularizer=t
f.keras.regularizers.l2(0.01), name=b+'skip_conv')(X_Skip)
    X_Skip = BatchNormalization(name=b + 'batch_norm_skip_conv')(X_Skip)

    # block_c + skip_conv
    X = Add(name = b+'add')([X , X_Skip])
    X = ReLU(name = b +'relu')(X)
    return X
```

Figure 3.5: The Convolution Block

The CONV_BLOCK is a key building block of the network, combining batch normalization, activation functions, and convolutional layers. It includes:

Residual Connection- The input is added to the block's output after processing, helping with deep network training by mitigating the vanishing gradient issue.

Layers -

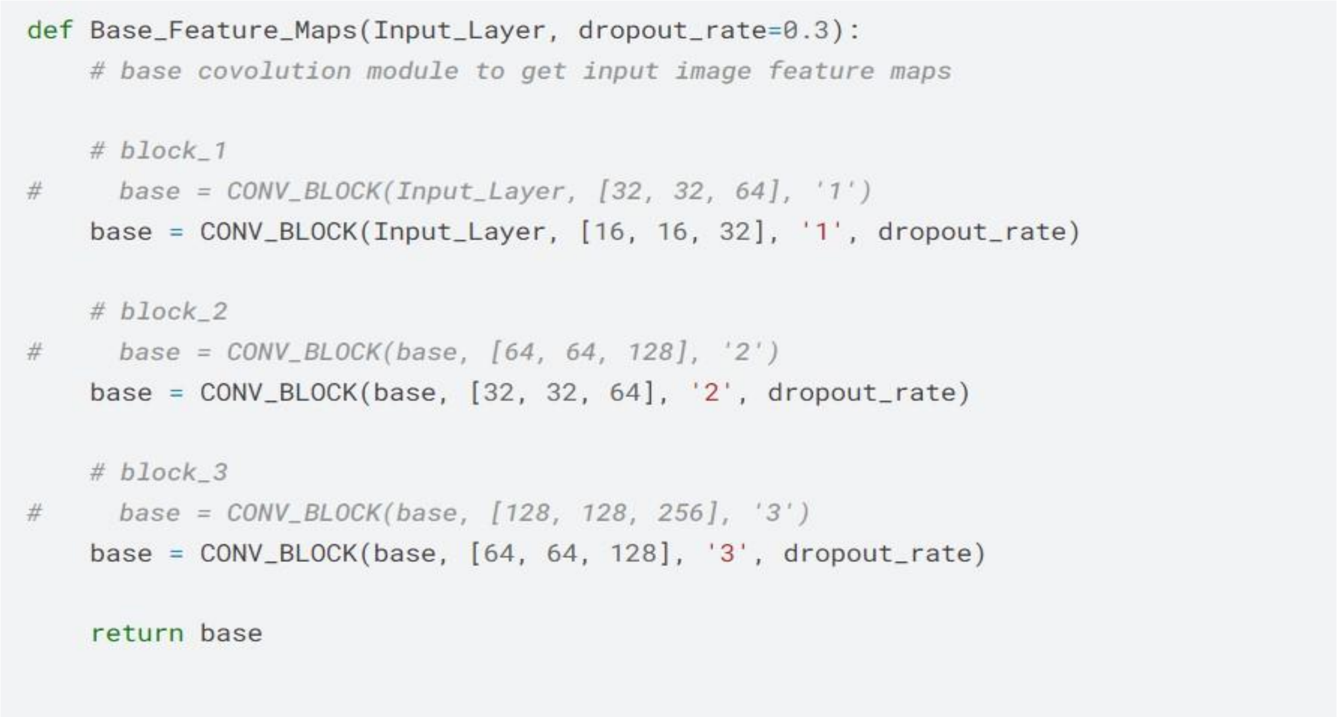
1x1 Convolution- Maintains or reduces the number of channels while keeping spatial dimensions unchanged.

3x3 Convolution with Dilation- Extracts information while preserving spatial resolution, increasing the receptive field without losing resolution.

1x1 Convolution- Adjust the output channels to match the input channels before adding the residual connection.

Skip Connection- A 1x1 convolution ensures the input channels match the output channels before addition.

B. The Base Feature Map

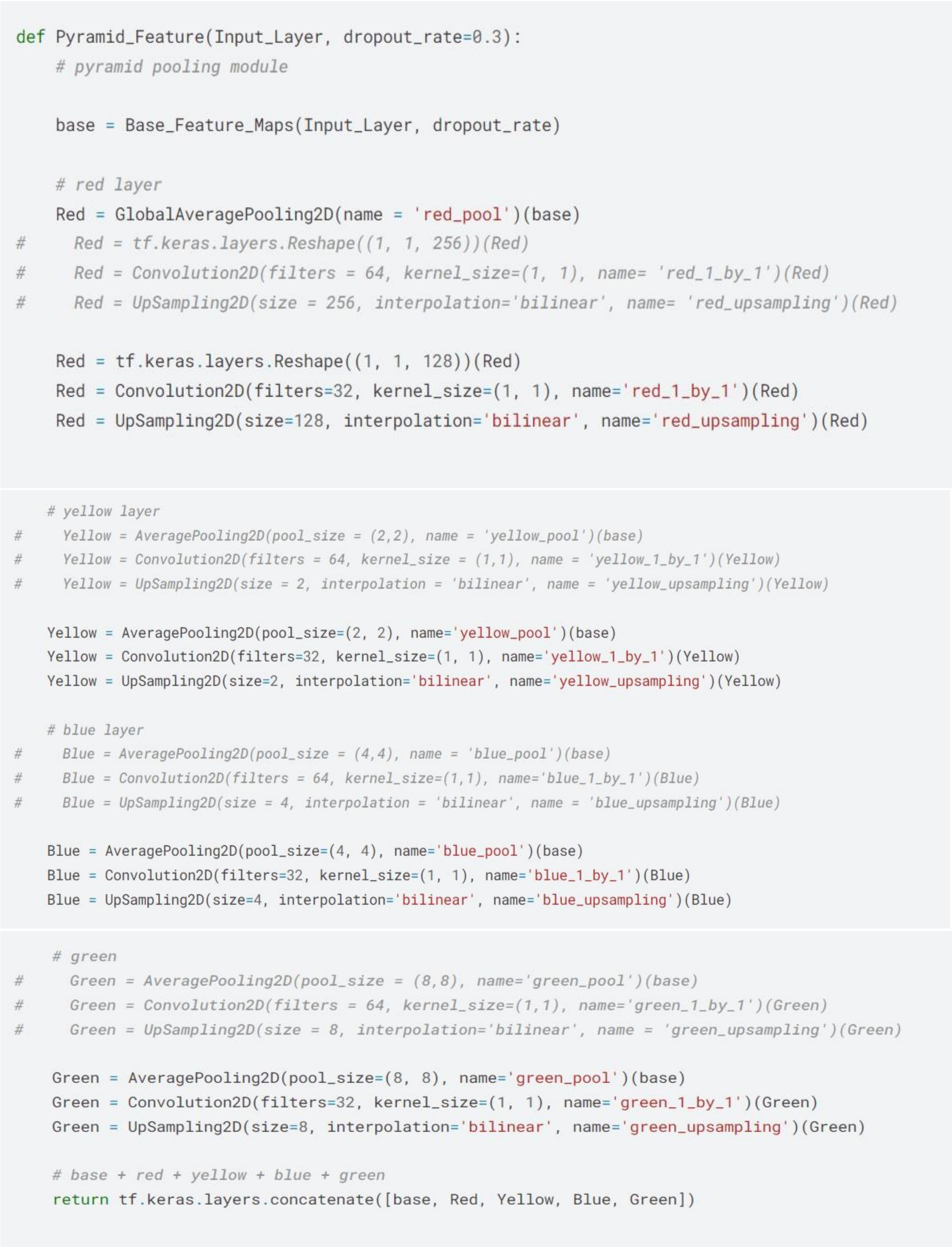


```
def Base_Feature_Maps(Input_Layer, dropout_rate=0.3):
    # base covolution module to get input image feature maps

    # block_1
#     base = CONV_BLOCK(Input_Layer, [32, 32, 64], '1')
    base = CONV_BLOCK(Input_Layer, [16, 16, 32], '1', dropout_rate)

    # block_2
#     base = CONV_BLOCK(base, [64, 64, 128], '2')
    base = CONV_BLOCK(base, [32, 32, 64], '2', dropout_rate)

    # block_3
#     base = CONV_BLOCK(base, [128, 128, 256], '3')
    base = CONV_BLOCK(base, [64, 64, 128], '3', dropout_rate)

    return base
```

Figure 3.6: The Base Feature Map

This network component is responsible for extracting features from the input image. It consists of multiple stacked **CONV_BLOCK** units. As the input passes through three consecutive convolution blocks, each with more filters than the previous one, feature maps are progressively extracted, allowing the network to capture more complex information for further processing.

C. The Pyramid Pooling Module (Pyramid Feature Map)



```
def Pyramid_Feature(Input_Layer, dropout_rate=0.3):
    # pyramid pooling module

    base = Base_Feature_Maps(Input_Layer, dropout_rate)

    # red layer
    Red = GlobalAveragePooling2D(name = 'red_pool')(base)
#     Red = tf.keras.layers.Reshape((1, 1, 256))(Red)
#     Red = Convolution2D(filters = 64, kernel_size=(1, 1), name= 'red_1_by_1')(Red)
#     Red = UpSampling2D(size = 256, interpolation='bilinear', name= 'red_upsampling')(Red)

    Red = tf.keras.layers.Reshape((1, 1, 128))(Red)
    Red = Convolution2D(filters=32, kernel_size=(1, 1), name='red_1_by_1')(Red)
    Red = UpSampling2D(size=128, interpolation='bilinear', name='red_upsampling')(Red)
```

```
    # yellow layer
#     Yellow = AveragePooling2D(pool_size = (2,2), name = 'yellow_pool')(base)
#     Yellow = Convolution2D(filters = 64, kernel_size = (1,1), name = 'yellow_1_by_1')(Yellow)
#     Yellow = UpSampling2D(size = 2, interpolation = 'bilinear', name = 'yellow_upsampling')(Yellow)

    Yellow = AveragePooling2D(pool_size=(2, 2), name='yellow_pool')(base)
    Yellow = Convolution2D(filters=32, kernel_size=(1, 1), name='yellow_1_by_1')(Yellow)
    Yellow = UpSampling2D(size=2, interpolation='bilinear', name='yellow_upsampling')(Yellow)

    # blue layer
#     Blue = AveragePooling2D(pool_size = (4,4), name = 'blue_pool')(base)
#     Blue = Convolution2D(filters = 64, kernel_size=(1,1), name='blue_1_by_1')(Blue)
#     Blue = UpSampling2D(size = 4, interpolation = 'bilinear', name = 'blue_upsampling')(Blue)

    Blue = AveragePooling2D(pool_size=(4, 4), name='blue_pool')(base)
    Blue = Convolution2D(filters=32, kernel_size=(1, 1), name='blue_1_by_1')(Blue)
    Blue = UpSampling2D(size=4, interpolation='bilinear', name='blue_upsampling')(Blue)
```

```
    # green
#     Green = AveragePooling2D(pool_size = (8,8), name='green_pool')(base)
#     Green = Convolution2D(filters = 64, kernel_size=(1,1), name='green_1_by_1')(Green)
#     Green = UpSampling2D(size = 8, interpolation='bilinear', name = 'green_upsampling')(Green)

    Green = AveragePooling2D(pool_size=(8, 8), name='green_pool')(base)
    Green = Convolution2D(filters=32, kernel_size=(1, 1), name='green_1_by_1')(Green)
    Green = UpSampling2D(size=8, interpolation='bilinear', name='green_upsampling')(Green)

    # base + red + yellow + blue + green
    return tf.keras.layers.concatenate([base, Red, Yellow, Blue, Green])
```

Figure 3.7: The Pyramid Pooling Layer

The key component of PSPNet is the Pyramid Pooling Module, which captures global contextual information through pooling at various scales:

Global Average Pooling (Red Layer)- Condenses the entire feature map into a single pixel to capture global context, then up-samples the feature map to its original size.

Multiple Scale Pooling-

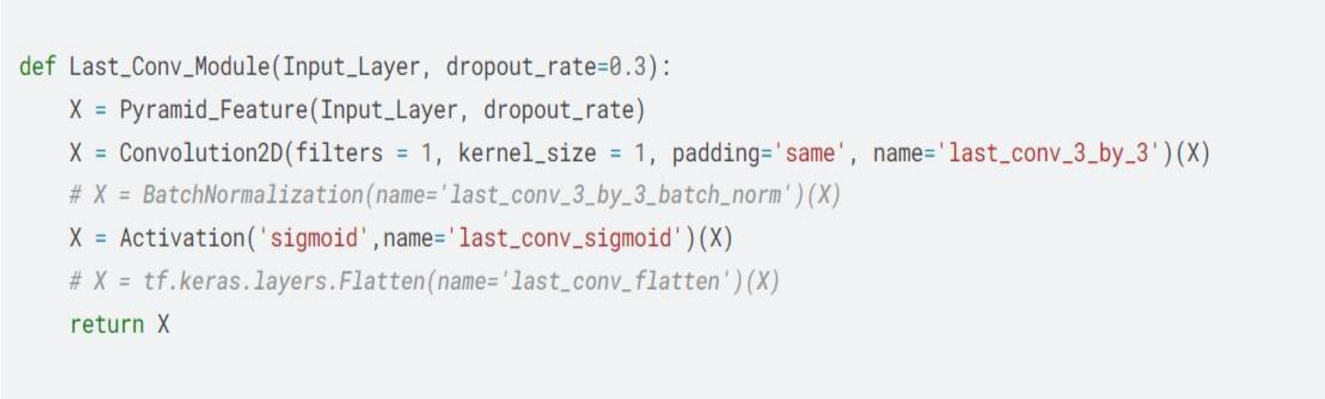
Yellow Layer- Uses 2x2 average pooling to capture medium-scale context, then up-samples to the original size.

Blue Layer- Applies 4x4 average pooling to capture a wider context, followed by upsampling.

Green Layer- Utilizes 8x8 average pooling for the largest context, then up-samples.

Concatenation- The output feature maps from the base layer and the Red, Yellow, Blue, and Green layers are concatenated along the channel dimension, blending local and global information for more accurate segmentation.

D. Final Convolution Layer



```
def Last_Conv_Module(Input_Layer, dropout_rate=0.3):
    X = Pyramid_Feature(Input_Layer, dropout_rate)
    X = Convolution2D(filters = 1, kernel_size = 1, padding='same', name='last_conv_3_by_3')(X)
    # X = BatchNormalization(name='last_conv_3_by_3_batch_norm')(X)
    X = Activation('sigmoid',name='last_conv_sigmoid')(X)
    # X = tf.keras.layers.Flatten(name='last_conv_flatten')(X)
    return X
```

Figure 3.8: The Last Convolution Layer

The concatenated features from the Pyramid Pooling Module are processed as follows-

- **1x1 Convolution**: Reduces the concatenated features to a single output feature map by applying a 1x1 convolution.
- **Sigmoid Activation**: The output is passed through a sigmoid activation function, generating a segmentation map where pixel values range from 0 to 1, representing the likelihood that each pixel belongs to the target class.

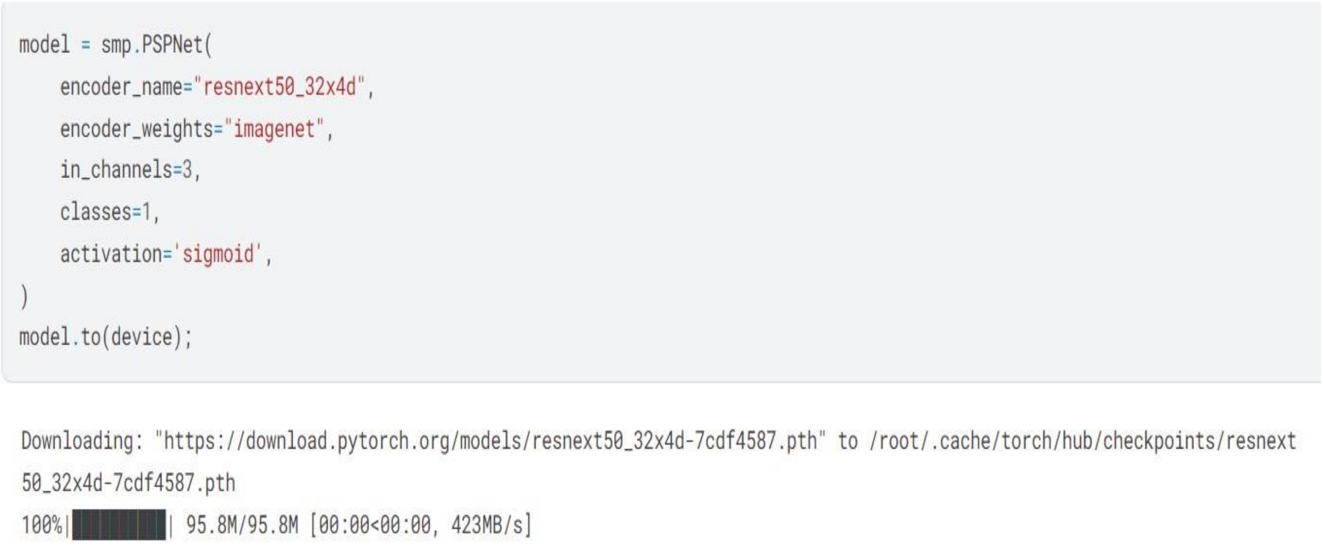
Advantages –

PSPNet provides more reliable predictions compared to other segmentation models due to its multi-scale feature extraction, which enables the extraction of both local and global information (Schumacher, 2023).

Disadvantages –

Training PSPNet from scratch is time-consuming, computationally expensive, and slow due to the depth of the layers, taking nearly 5 hours to complete. The model trained from scratch did not achieve good segmentation of brain tumors. In response to the research question in the Introduction, this scratch model had lower accuracy, Dice, and IoU scores in training, test, and validation datasets compared to the PSPNet transfer learning model and the UNet EfficientNetB7 transfer learning model, as will be discussed in the Results chapter.

3.3.1.2 PSPNet with Transfer Learning



```
model = smp.PSPNet(
    encoder_name="resnext50_32x4d",
    encoder_weights="imagenet",
    in_channels=3,
    classes=1,
    activation='sigmoid',
)
model.to(device);
```

```
Downloading: "https://download.pytorch.org/models/resnext50_32x4d-7cdf4587.pth" to /root/.cache/torch/hub/checkpoints/resnext
50_32x4d-7cdf4587.pth
100%|██████████| 95.8M/95.8M [00:00<00:00, 423MB/s]
```

Figure 3.9: Transfer Learning Model Of PSPNet with ResNeXt50-32x4d Encoder

PSPNet - A deep learning model specifically tailored for semantic segmentation tasks is the Pyramid Scene Parsing Network (PSPNet) (Schumacher, 2023).

ResNeXt50-32x4d - ResNeXt50-32x4d is a variant of the ResNet model with a unique feature map aggregation architecture used as the encoder (backbone). In the "32x4d" configuration, "32" represents the number of groups, and "4d" refers to the width of each group in the model's architecture (Xie et al., 2016).

Transfer Learning- The model is initialized with pre-trained weights from the ImageNet dataset (encoder_weights="imagenet"), meaning the encoder has already learned to extract general features from images. This allows it to be fine-tuned for the specific task of MRI segmentation.

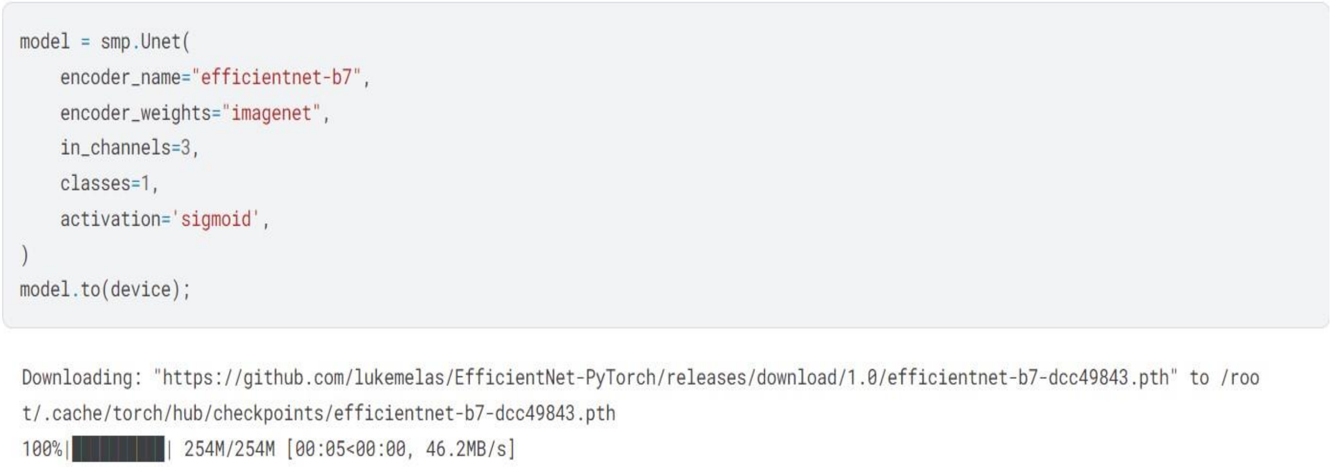
Advantages -

The PSPNet model with a ResNeXt50-32x4d encoder, pre-trained on ImageNet, is faster to train compared to the version trained from scratch. It performs well on metrics like accuracy, dice score, IoU score, and loss. Since the encoder already has useful features for pattern recognition, the model converges more quickly. As per the research question 1 this PSPNet with transfer learning model has achieved better accuracy, dice and IoU score in training, test and validation dataset compared to the model from scratch.

For research question 2, the PSPNet model with transfer learning significantly reduces training time and achieves better segmentation compared to the PSPNet model trained from scratch. The combination of faster training and more accurate brain tumor segmentation gives the transfer learning model a clear advantage over the scratch model.

3.3.2 UNet-EfficientNetB7 Transfer Learning Model

As mentioned in the beginning of the Modelling Section, UNet EfficientNetB7 combines EfficientNetB7's strong feature extraction with UNet's proven effectiveness in biomedical image segmentation. Both models were trained on the same dataset with hyperparameters tailored to each architecture.



```
model = smp.Unet(
    encoder_name="efficientnet-b7",
    encoder_weights="imagenet",
    in_channels=3,
    classes=1,
    activation='sigmoid',
)
model.to(device);
```

```
Downloading: "https://github.com/lukemelas/EfficientNet-PyTorch/releases/download/1.0/efficientnet-b7-dcc49843.pth" to /roo
t/.cache/torch/hub/checkpoints/efficientnet-b7-dcc49843.pth
100%|██████████| 254M/254M [00:05<00:00, 46.2MB/s]
```

Figure 3.10: Transfer Learning Model Of UNet with EfficientNetB7 Encoder

UNet- Another well-liked design for segmenting images is U-Net, which has a U-shaped structure that allows it to collect both low- and high-level characteristics. Ronneberger, Fischer and Brox (2015) presented this model for effective biomedical image segmentation.

EfficientNet-B7 is an encoder that balances performance and resource usage. It is highly efficient and scales both in depth and width to improve model performance. Tan and Le (2019) first introduced the EfficientNet model.

Transfer Learning- The model's encoder, pre-trained on the ImageNet dataset (encoder_weights = "imagenet"), allows it to start with a strong set of features. These features can then be fine-tuned for the specific segmentation task, improving its performance in image analysis.

Advantages –

The UNet model with EfficientNetB7 model is faster and computationally less expensive. Performed well in case of performance metrics evaluation.

Similar to the PSPNet Transfer Learning model, the UNet EfficientNetB7 model has outperformed the scratch model in terms of accuracy, dice score, and IoU score across the training, test, and validation datasets, as indicated by research question 1.

The UNet EfficientNetB7 model, like the PSPNet transfer learning model, takes less training time compared to a model trained from scratch. It also provides more accurate segmentation than the PSPNet scratch model. As a result, the UNet EfficientNetB7 model has a significant advantage over the scratch model due to its faster training time and accurate brain tumor segmentation, as highlighted in research question 2.

3.4 Performance Evaluation

This section presents the performance metrics used to evaluate the brain MRI segmentation models, including accuracy, F1 score, loss function, Dice coefficient, and Intersection over Union (IoU) score. These metrics provide a comprehensive assessment of the model's segmentation quality, overlap, and accuracy.

All the metrics have been discussed below thoroughly.

3.4.1 F1 Score

The F1 score measures the harmonic mean of precision and recall. It combines accuracy and recall into a single metric, providing a better understanding of model performance. The F1 score is commonly used for evaluating binary and multi-class classification (Sharma, 2024).

Depending on your use case, you can change the F-score to F0.5, F1, or F2 based on how much weight is given to precision over recall (Sharma, 2024).

According to Kundu (2024), To understand how the F1 score is determined, a confusion matrix must be examined. It represents a model's prediction ability on a dataset. For a binary class dataset, a confusion matrix consists of four key elements (let's say that it contains "positive" and "negative" classes)-

- True Positives (TP)- Samples that were accurately predicted to be "positive".
- False Positives (FP)- Count of samples that were incorrectly identified as "positive."
- True Negatives (TN)- Number of samples properly predicted as "negative".
- False Negatives (FN)- Count of samples that were incorrectly identified as "negative"

Precision - Precision, also known as Positive Predictive Value (PPV), is a model evaluation metric that represents the fraction of predicted positive values that are actually correct. It measures how accurate the model is when predicting the positive class (Sharma, 2024).

The precision is calculated in the way –

$$\mathrm{Pr}\,e\,cision = TP/(TP + FP)$$

Figure 3.11: Formula of Precision

Recall - Recall is a performance metric that evaluates a model's ability to identify the positive class. It measures the percentage of actual positive values, including false negatives, that are correctly predicted as positive (Sharma, 2024).

The Recall is calculated as below –

$$Recall = TP/(TP + FN)$$

Figure 3.12: Formula of Recall

F1 Score - The F1 score uses a harmonic mean to proportionally reflect recall and precision (Sharma, 2024).

The F1 score is calculated as below –

$$F1\,Score = (2 * \Pr e\,cision * Recall)/(\Pr e\,cision + Recall)$$

Figure 3.13: Formula of F1 Score

3.4.2 Accuracy –

The accuracy metric measures a model's overall performance across all classes and is especially useful when all classes are equally important. It is calculated by dividing the number of correct predictions by the total number of predictions (Ahmed Fawzy Gad, 2020).

$$Accuracy = (TP + TN)/(TP + TN + FP + FN)$$

Figure 3.14: Formula For Accuracy

3.4.3 Loss Function –

The loss function, also known as the error function, is an essential component in machine learning. It measures the difference between the target values and the predicted outputs of an algorithm, guiding model optimization (Richmond Alake, 2023).

There are different types of loss functions used in Classification and Regression problems.

3.4.3.1 Loss Function for Regression –

For regression problems, MSE (Mean Squared Error) and MAE (Mean Absolute Error) are used most of the time.

MSE Loss or L2 Loss–

Mean Squared Error (MSE), also known as L2 loss, is a loss function that calculates the average of the squared differences between predictions and target values. It penalizes larger deviations more heavily by squaring the errors, providing a greater penalty for larger errors. The mean of these squared errors is then normalized by the number of samples in the dataset or observations (Richmond Alake, 2023).

MAE Loss or L1 Loss –

Mean Absolute Error (MAE), also known as L1 Loss, is a loss function used in regression tasks that calculates the average of the absolute differences between predicted and target values. Unlike Mean Squared Error (MSE), which squares the differences, MAE treats all errors equally, regardless of their magnitude (Richmond Alake, 2023).

3.4.3.2 Loss Function for Classification –

For a classification problem, Binary Cross Entropy and Categorical Loss Entropy are used most of the time.

Binary Cross Entropy Loss/ Log Loss –

Binary Cross-Entropy Loss (BCE) is a performance metric for classification models that predicts a probability value between 0 and 1, representing the likelihood that a data sample belongs to a particular class. BCE is used for binary classification with two classes. For multiclass classification, a variation called Categorical Cross-Entropy Loss is applied (Richmond Alake, 2023).

3.4.4 Dice Score / Coefficient

The Dice coefficient is the harmonic mean of recall and precision, calculated as 2 times the intersection divided by the total pixels in both images. It measures similarity between two sets. (Huynh, 2023).

$$Dice = \frac{2TP}{2TP + FP + FN}$$

Figure 3.15: Dice Coefficient

3.4.5 IoU (Intersection Over Union) Coefficient

The intersection over union (IoU), or Jaccard index, is the area where the anticipated segmentation and the ground truth intersect (Huynh, 2023).

$$IOU = \frac{TP}{TP + FP + FN}$$

Figure 3.16: IoU Coefficient

3.5 Summary –

In addition to the performance evaluation criteria included in this study, the methods chapter covers data collection, preprocessing, and modeling. The upcoming chapter will present the research results along with a critical analysis and evaluation of the findings.

Chapter 4 Result

This chapter discusses the results of three models - PSPNet from Scratch, PSPNet with Transfer Learning, and UNet with EfficientNetB7 Transfer Learning—and provides a comparative analysis of the Brain Tumor Segmentation models.

4.1 Results Acquired By PSPNet From Scratch –

4.1.1 Brain MRI Images Along with Mask Images

The image below is the brain MRI image along with the mask image.

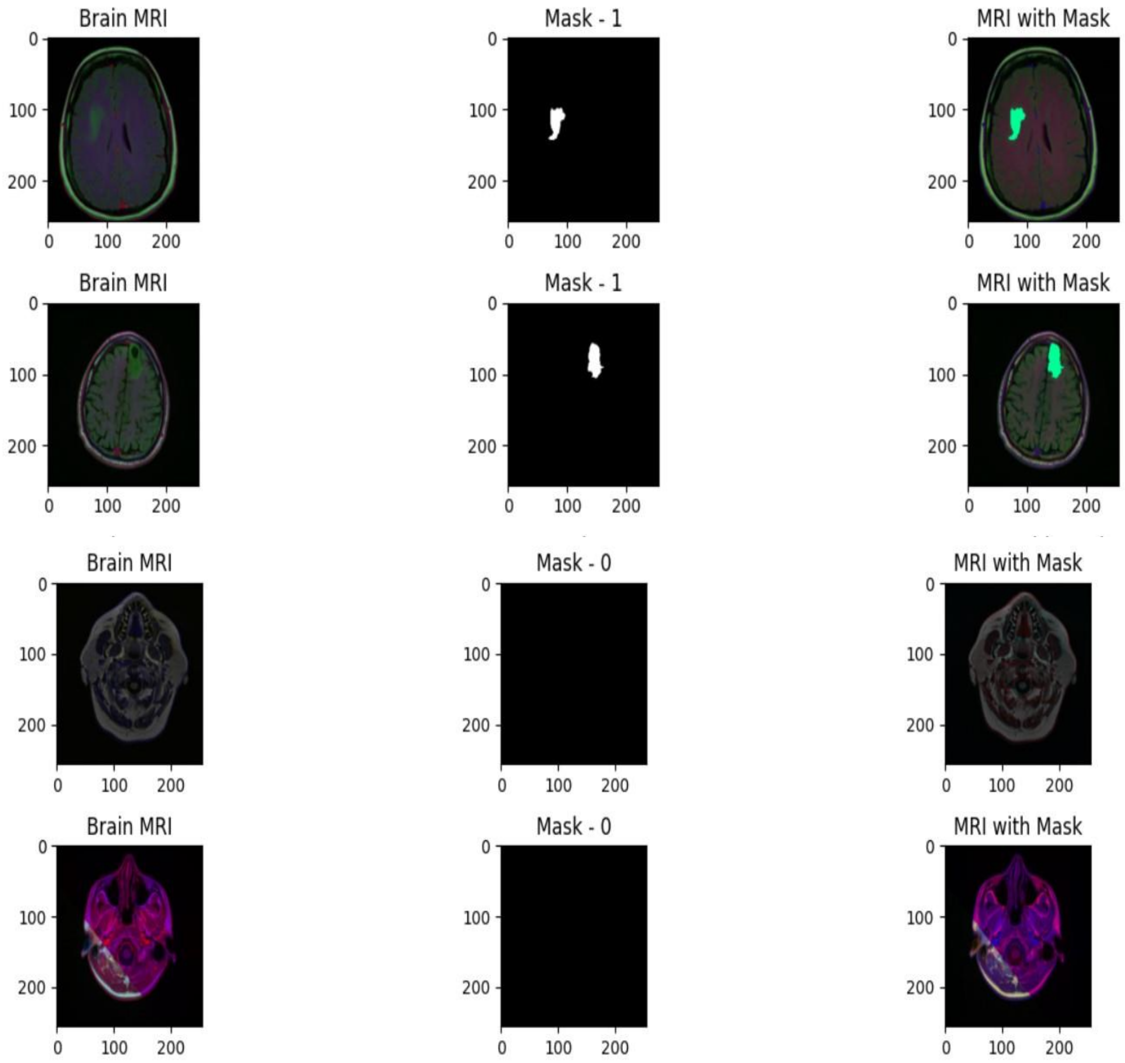


Figure 4.1: Brain MRI Images along with Mask Images

The image shows three columns: Brain MRI images, Mask Images, and MRI with Mask, which will later be predicted by the PSPNet model.

Let us discuss every column.

1. Brain MRI - These raw brain MRI scans are displayed in the leftmost column. With each image, a conventional MRI color scheme is used to depict a slice of a brain.

2. Mask Image –

Mask 1 – The middle column displays segmentation masks produced by the model to identify regions of interest in brain MRIs, such as tumors. The white area in the mask indicates the region detected as an abnormality or important area, like a tumor.

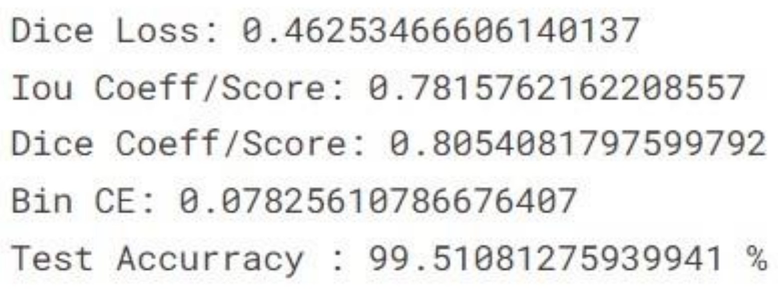
Mask 0 - In this case, the segmentation masks are entirely black, indicating the model did not detect any regions of interest. The absence of white areas suggests the model predicts no abnormalities or significant observations in these specific slices.

3. MRI with Mask –

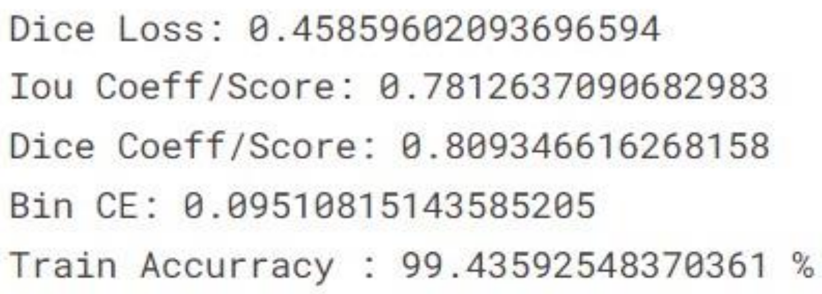
Mask 1 - The final column shows the superimposition of the segmentation mask onto the original brain MRI. The green area represents the mask, overlaid on the corresponding MRI slice, highlighting where the model predicts an abnormality or lesion.

Mask 0 - The final column shows the mask superimposed on the original MRI images. Since the mask is black, no green areas are visible, indicating that the model did not detect any areas of concern.

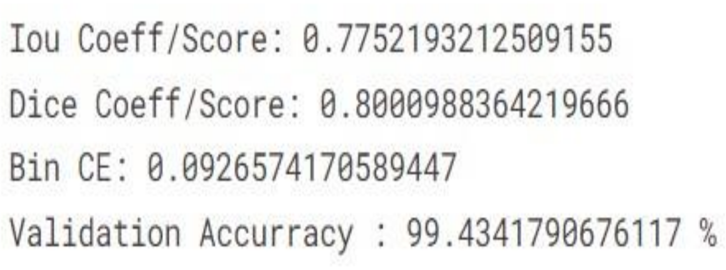
4.1.2 The Results on Train, Test and Validation Data



```
Dice Loss: 0.46253466606140137
Iou Coeff/Score: 0.7815762162208557
Dice Coeff/Score: 0.8054081797599792
Bin CE: 0.07825610786676407
Test Accurracy : 99.51081275939941 %
```



```
Dice Loss: 0.45859602093696594
Iou Coeff/Score: 0.7812637090682983
Dice Coeff/Score: 0.809346616268158
Bin CE: 0.09510815143585205
Train Accurracy : 99.43592548370361 %
```



```
Iou Coeff/Score: 0.7752193212509155
Dice Coeff/Score: 0.8000988364219666
Bin CE: 0.0926574170589447
Validation Accurracy : 99.4341790676117 %
```

Figure 4.2: Dice Score, IOU Score, Accuracy, and Binary Cross Entropy Value acquired By PSPNet (scratch) in Three different dataset

The image above describes performance metrics, including Dice score, IOU, Accuracy, and Binary Cross Entropy, for three datasets: Test, Train, and Validation.

Test Dataset –

The model shows exceptional performance on the test set with 99.51% accuracy, indicating it accurately predicts most pixels. Although the Dice Loss is around 0.46, the high Dice Coefficient and IOU score suggest effective identification of regions of interest. The low binary cross-entropy (0.0783) indicates minimal classification errors.

Train Dataset –

The training performance closely mirrors the test performance, with a Dice Loss of 0.4586 and 99.44% accuracy. The IOU and Dice Coefficients show a slight improvement on the training data, as expected since the model was trained on this set. The Binary Cross-Entropy (Bin CE) is slightly higher at 0.0951, indicating marginally more prediction errors compared to the test set, but the difference is minimal overall.

Validation Dataset –

The validation performance is similar to the training performance, with a validation accuracy of 99.43%. While the IOU and Dice Coefficients are slightly lower than on the training set, they remain high, showing the model's effectiveness on unseen data. The Binary Cross-Entropy (Bin CE) is 0.0927, closely aligning with the training set, indicating consistent performance between training and validation.

Overall Comparison –

Consistency Across Datasets- The model demonstrates consistent performance across all three datasets (test, train, and validation), achieving an accuracy of around 99.4%-99.5% in each case. The IOU and Dice Scores are also highly similar across the datasets, with only minimal differences.

Generalization- The model exhibits strong generalization to new, unseen data in the test and validation sets. There are only minor discrepancies between the training and validation/test scores, with no signs of significant overfitting, as the training metrics closely match the test and validation results.

4.1.3 Predicted Images (Tumor Areas)

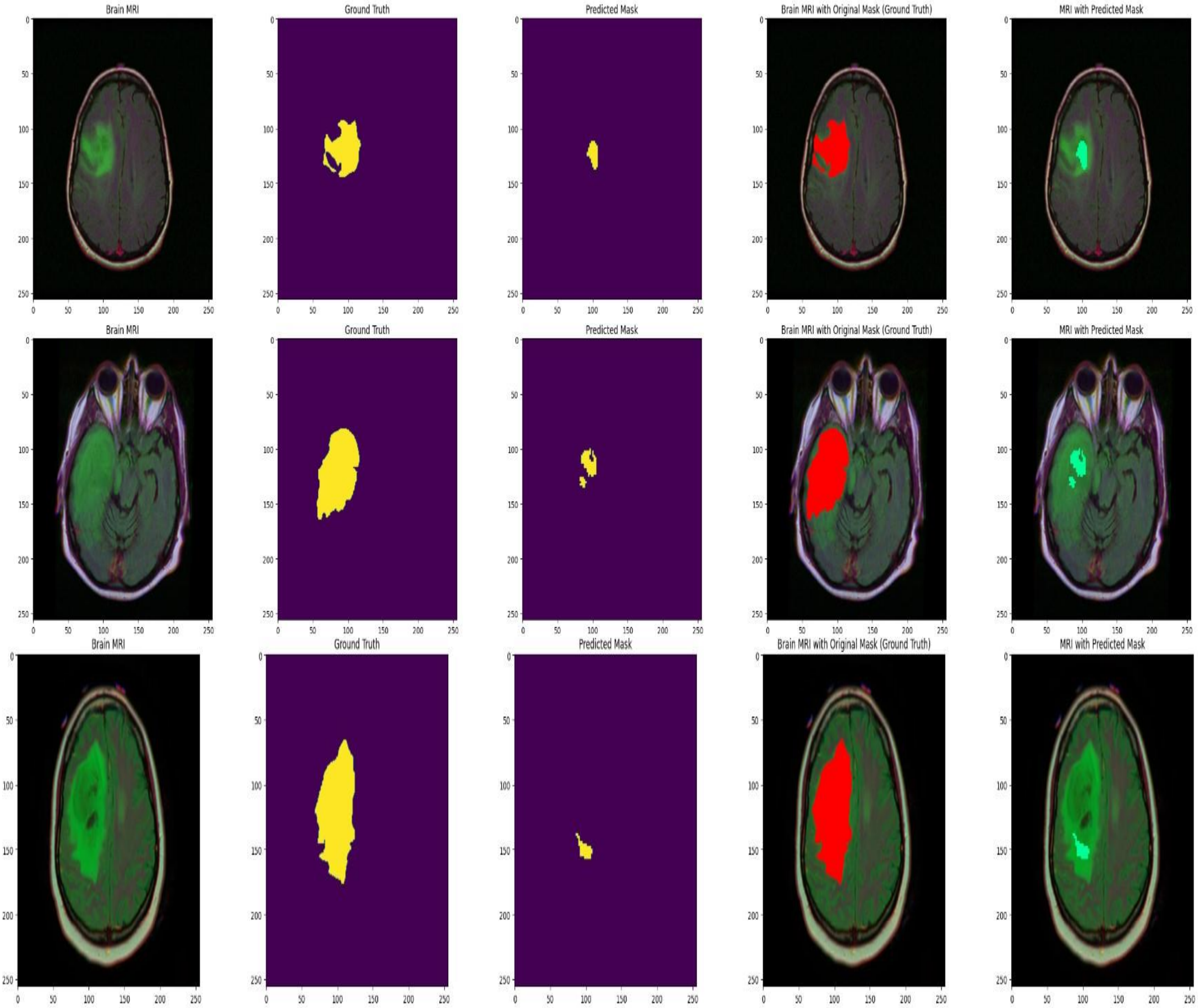


Figure 4.3: Tumor Region Predicted by PSPNet (Scratch)

Analysis of Each Column –

Brain MRI- Displays slices of the brain exhibiting visible abnormalities that the model is intended to detect.

Ground Truth- The yellow areas on the ground truth mask indicate regions of interest, such as tumors. These regions are moderately large and well defined**.**

Predicted Mask- The model's predictions show limited overlap with the ground truth mask, indicating that the PSPNet model is making conservative predictions and not fully capturing the entire region of interest.

The MRI with the original mask (ground truth)- demonstrates the alignment between the actual ground truth and the brain MRI. The red region delineates the actual area of interest.

MRI with Predicted Mask- The model's prediction, shown by the green overlay, does not fully capture the abnormality. This under-prediction suggests the model is missing a significant portion of the area it should be detecting.

From the above figure it could be observed that –

The Ground Truth Masks (yellow)- are significantly more extensive in delineating the areas of interest compared to the Predicted Masks (yellow) produced using the PSPNet model.

Predicted Mask Precision- The predictions, while not perfect, can identify certain parts of the lesion but fail to cover the full extent of the abnormalities. This suggests that the model has learned to recognize some key features but needs further adjustment or training to improve its prediction accuracy.

Under-Prediction- The model's predicted masks are smaller than the actual ground truth masks, indicating inaccurate segmentation. This issue is common in early-stage or under-trained models, where the model struggles to detect the full extent of anomalies.

Overall, from the Model PSPNet from Scratch, It can be said that –

The PSPNet model identifies some regions of interest but inaccurately estimates the actual area, leading to inconsistencies between the predicted and actual masks. It misses key areas of interest.

4.1.4 Visualization On Performance Metrics –

The visualizations are described here for performance metrics obtained by the PSPNet Model.

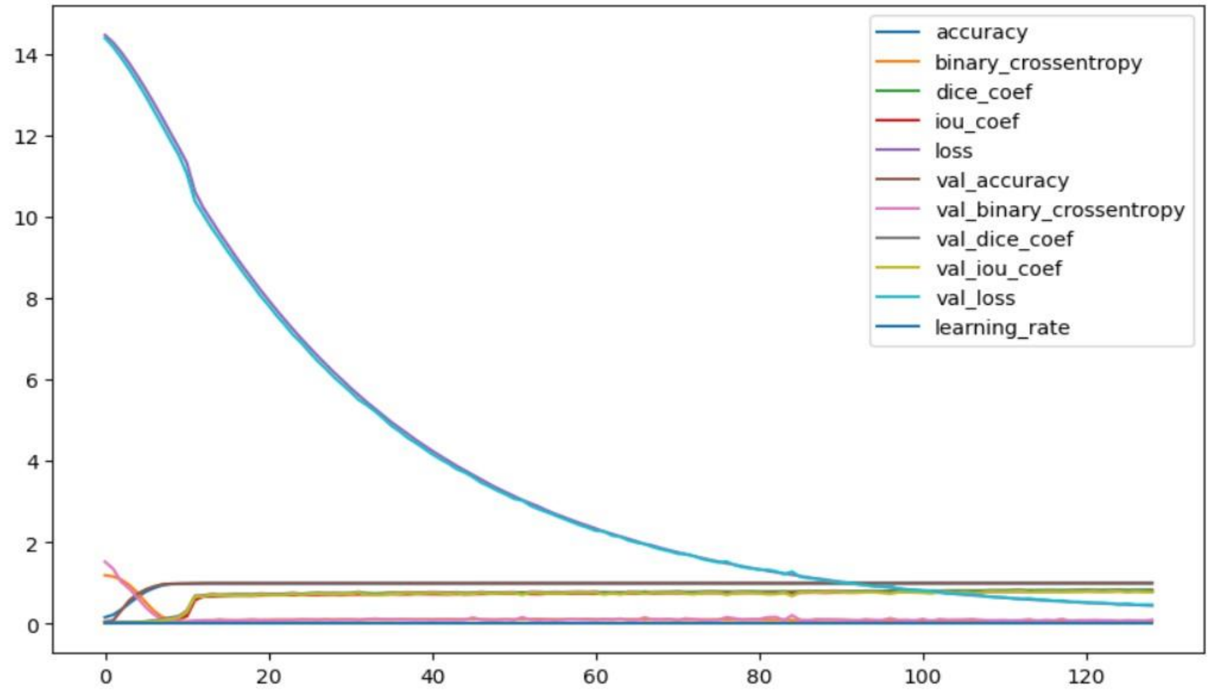


Figure 4.4: Visualization on Accuracy, Binary Crossentropy, accuracies

The accuracy (training and validation) - The accuracy (training in blue and validation in brown) reflects the model's ability to predict pixel classes accurately. The consistent and high accuracy throughout training indicates the model maintains strong performance as the epochs progress.

Binary Cross-Entropy Loss- shown in orange for training and pink for validation, evaluates the model's pixel classification performance. The decreasing loss over time indicates effective learning and improved predictive capabilities. However, the faster drop in training loss compared to validation loss suggests potential overfitting, highlighting the need for measures to manage it.

The Dice coefficient- calculated for both training (green) and validation (black), measures the overlap between predicted and ground truth masks. A higher Dice coefficient indicates better segmentation. The data shows consistent improvement, eventually stabilizing, suggesting the model is effectively learning to segment regions accurately.

The IOU Coefficient- measured for both training (red) and validation (yellow), evaluates the model's accuracy in predicting correct regions within an image. Like the Dice coefficient, a higher value indicates better performance. The graph shows steady progress over time for both training and validation.

The learning rate, shown by the light blue line, remains constant throughout training. While a fixed learning rate can be effective initially, adjusting it, especially by decreasing it in later epochs could improve fine-tuning and enhance the model's performance.

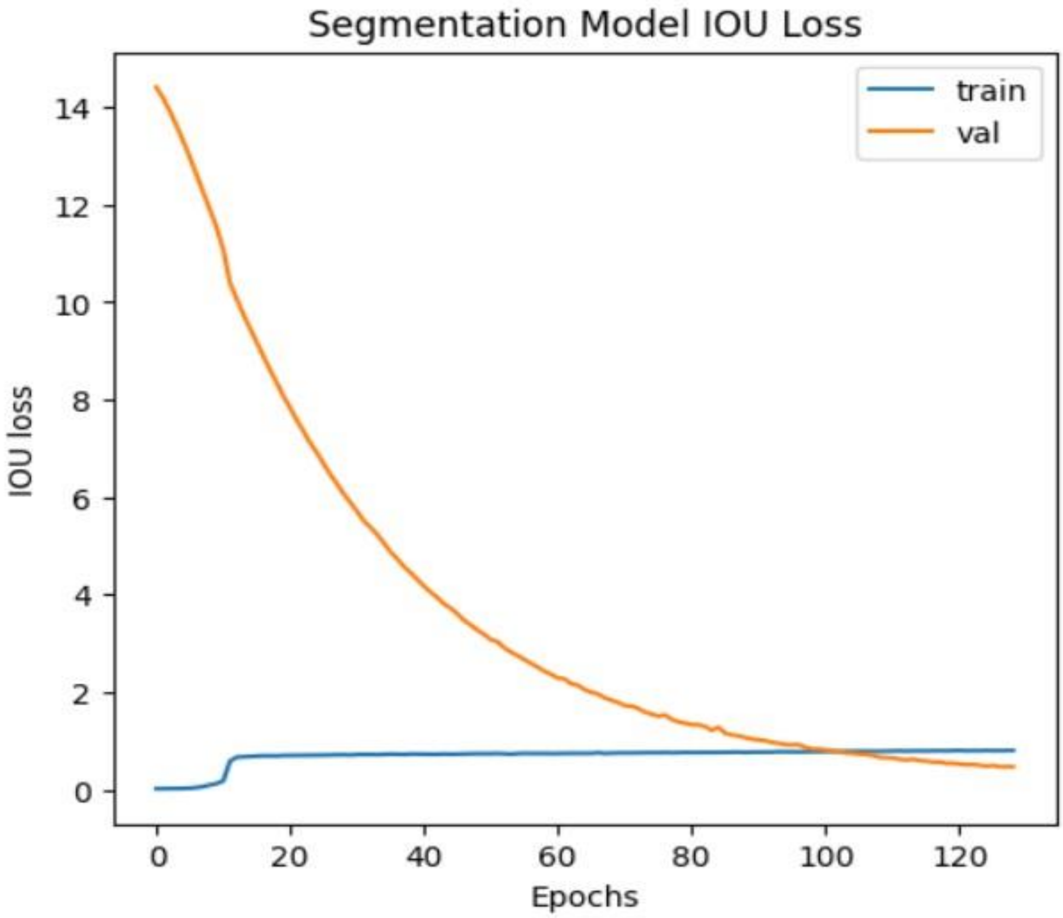


Figure 4.5: Epoch Vs IOU Loss (Training and Validation)

This graph analyzes IOU loss for both the training and validation datasets over the epochs. IOU loss measures the overlap between predicted segmentation and ground truth.

The training IOU loss (blue line) remains consistently low with minimal variation, indicating rapid convergence and strong performance early in training.

The validation IOU loss (orange line) starts high, reflecting poor performance on unfamiliar data. However, it steadily decreases as training progresses, converging towards the training loss. This suggests the model is learning to generalize effectively to new data, with the sharp drop in validation loss indicating significant improvement.

4.2 Results Acquired by PSPNet with Transfer Learning Model and UNet EfficuientNetB7 Model–

4.2.1 MRI Images and Mask Images –

The image below is actually MRI Images with Abnormality Mask Images

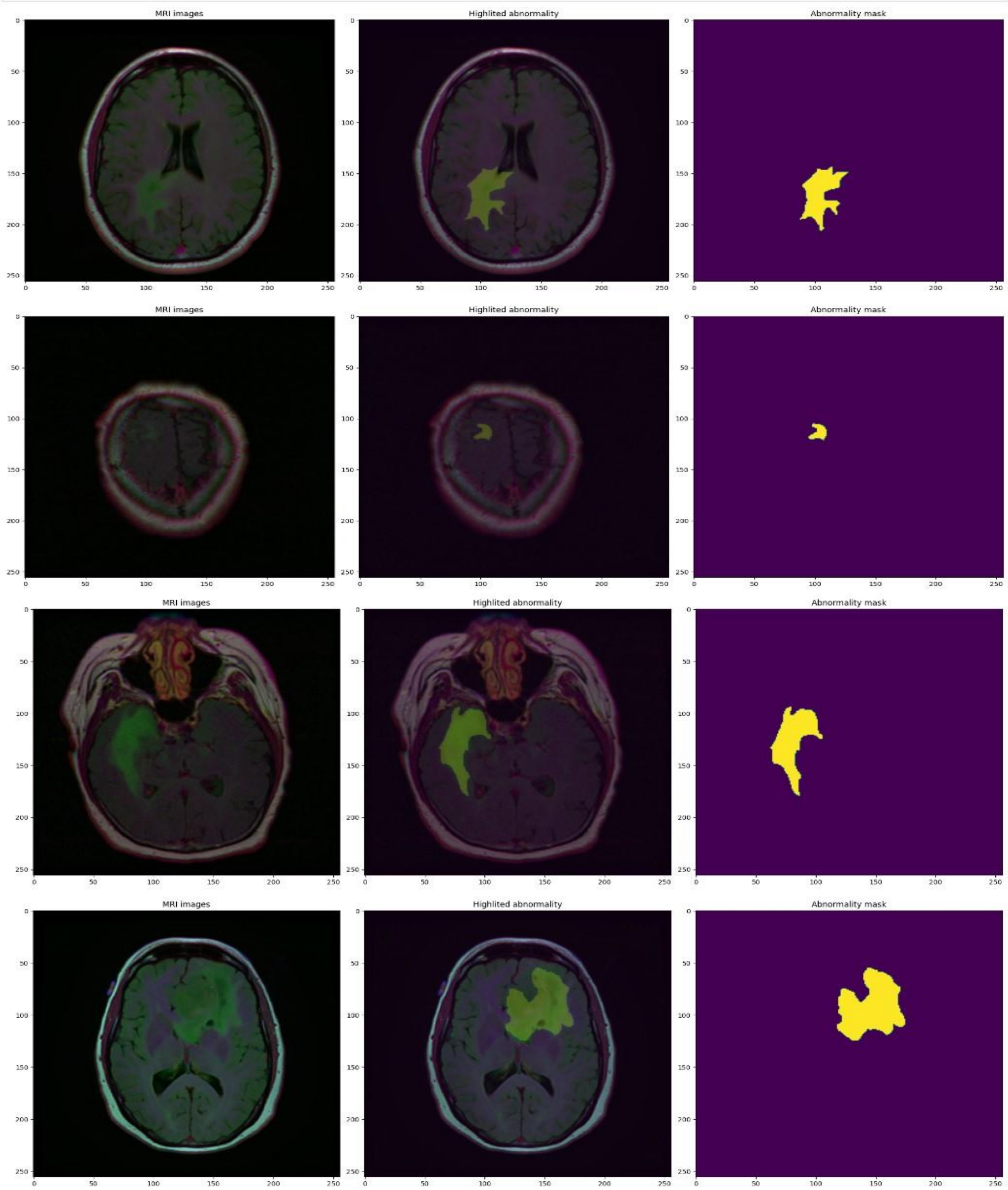


Figure 4.6: MRI Image with Abnormality Mask

Let’s discuss every column first.

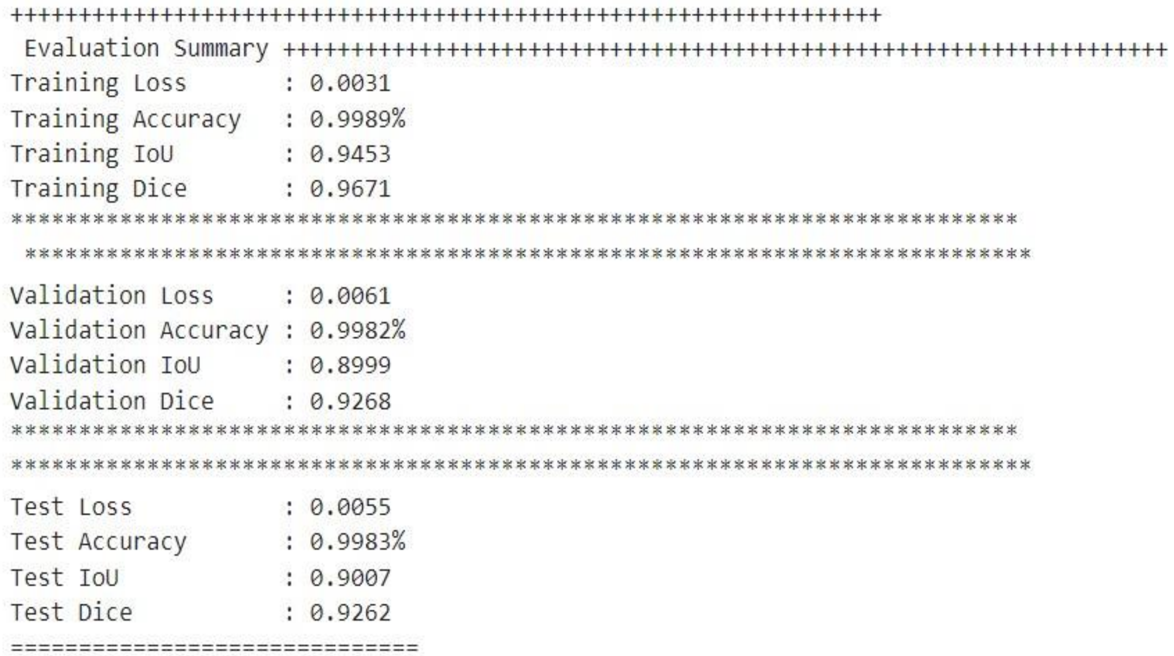
MRI Images- This column shows the original brain MRI slices, with potential green highlights indicating pre-processed enhancements to emphasize abnormalities.

Highlighted Abnormality- This column overlays the predicted abnormalities on the MRI images, with the green area indicating the region of interest (e.g., a tumor), helping visualize where the model detects potential abnormalities.

Abnormality Mask- This column displays segmentation masks, with the yellow area representing the model's predicted abnormal region, making it easier to compare the model's detection with the actual data.

4.2.2 Results on Train Test and Validation Data

The below figure shows Performance Metrics for UNet with EfficientNetB7 model-



```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
 Evaluation Summary +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Training Loss       : 0.0031
Training Accuracy   : 0.9989%
Training IoU        : 0.9453
Training Dice       : 0.9671
**************************************************************************
 **************************************************************************

Validation Loss     : 0.0061
Validation Accuracy : 0.9982%
Validation IoU      : 0.8999
Validation Dice     : 0.9268
**************************************************************************
**************************************************************************

Test Loss           : 0.0055
Test Accuracy       : 0.9983%
Test IoU            : 0.9007
Test Dice           : 0.9262
==============================
```

Figure 4.7: Performance Metrics(IOU, DICE, Loss, and Accuracy) Results for UNet EfficientNetB7 on Train Validation and Test Dataset

The above Image delineates the performance metrics results we have got from UNet EfficientNetB7 model.

1. Training Performance -

Training Loss - 0.0031.

The low training loss suggests that the model effectively minimized errors on the training set, indicating good convergence during training.

Training Accuracy - 0.9989%

The model demonstrates exceptionally high accuracy on the training set, correctly predicting nearly all pixels for their respective classes.

Training IoU - 0.9453.

The IoU measures how well the predicted segmentation aligns with the actual segmentation (ground truth). A value of 0.9453 is very high, indicating significant overlap between the predicted and actual regions.

Training Dice Coefficient- 0.9671.

The Dice coefficient, which is more sensitive to smaller regions, measures overlap. A score of 0.9671 indicates excellent performance in detecting and segmenting regions of interest, such as abnormalities.

2. Validation Performance –

Validation loss- 0.0061.

The validation loss is slightly higher than the training loss, as expected, indicating that the model generalizes well. However, there may still be room for improvement in overall performance.

Validation accuracy - 0.9982%

The validation accuracy is exceptionally high, nearing 100%, indicating strong generalization of the model to unseen data.

Validation IoU - 0.8999

The IoU for the validation set is slightly lower than the training IoU, which is typical for a well-trained model. A result of 0.8999 is still very high, indicating strong performance.

Validation Dice coefficient: 0.9268.

The Dice coefficient for the validation set remains high, indicating that the model continues to perform well in segmenting regions on unseen data.

3. Test Performance -

Test Loss- 0.0055.

The test loss is similar to the validation loss, indicating consistent performance. The slightly lower test loss suggests the model may perform even better on this particular test set.

Test Accuracy- 0.9983%.

The model's accuracy on the test set is extremely high, comparable to the training and validation accuracy, indicating excellent performance and strong generalization.

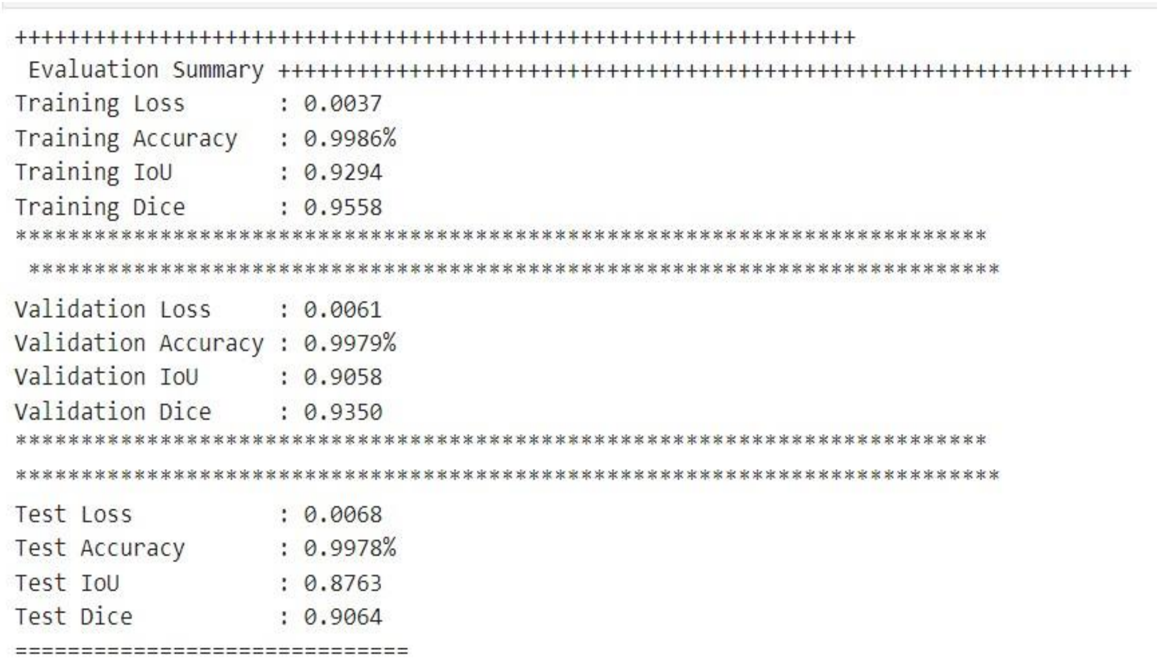
Test IoU- 0.9007

The IoU on the test set is slightly higher than on the validation set, indicating that the model performs well on new data and predicts segmentation zones with significant overlap with the ground truth.

Test Dice Coefficient- 0.9262.

Similar to the IoU, the Dice coefficient remains high, indicating that the model accurately segments both large and small regions on the test set.

The below figure shows the performance metrics of the PSPNet Transfer Learning Model



```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
 Evaluation Summary ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Training Loss       : 0.0037
Training Accuracy   : 0.9986%
Training IoU        : 0.9294
Training Dice       : 0.9558
**************************************************************************
 **************************************************************************
Validation Loss     : 0.0061
Validation Accuracy : 0.9979%
Validation IoU      : 0.9058
Validation Dice     : 0.9350
**************************************************************************
***************************************************************************
Test Loss           : 0.0068
Test Accuracy       : 0.9978%
Test IoU            : 0.8763
Test Dice           : 0.9064
==============================
```

Figure 4.8: Performance Metrics (IOU, DICE, Loss, and Accuracy) Results for PSPNet on Train Validation and Test Dataset

1. Training Performance -

Training Loss- 0.0037

The extremely low training loss indicates that the model minimizes errors well during training, showing that it has effectively learned the segmentation task.

Training Accuracy- 0.9986 percent

A training accuracy of 99.86% indicates that the model correctly predicts the class for nearly all pixels in the training set, reflecting strong performance during training.

Training IoU - 0.9294.

The IoU score of 0.9294 indicates a strong overlap between the predicted and actual ground truth segmentations, demonstrating the model's effectiveness in detecting areas of interest, such as tumors or lesions.

Training Dice Coefficient- 0.9558

The Dice coefficient of 0.9558 indicates an excellent match between the predicted segmentation and the ground truth, reflecting a high degree of overlap. This shows the model performs effectively, even in smaller or more challenging areas of the training data.

2. Validation Performance –

Validation loss- 0.0061

The validation loss is slightly higher than the training loss, which is expected. This indicates that the model generalizes well, though there is a minor difference in performance when compared to the training data.

Validation accuracy- 0.9979 %

The validation accuracy of 99.79% is very high, indicating that the model performs well on the validation data and generalizes effectively.

Validation IoU - 0.9058.

The validation IoU of 0.9058 is slightly lower than the training IoU but still very good. This suggests that the model continues to perform well, though the minor drop in IoU may indicate slight overfitting to the training set.

Validation Dice Coefficient- 0.9350

The validation Dice coefficient of 0.9350 is also high, indicating that the model's predictions remain closely aligned with the actual segmentations in the validation set. However, the slight drop compared to the training Dice score suggests a minor decline in performance.

3. Test Performance –

Test Loss- 0.0068

The test loss is slightly higher than the validation and training losses, as expected, since the model does not see the test set during training. This slight increase suggests that the model generalizes fairly well.

Test accuracy- 0.9978%.

The test accuracy of 99.78% shows that the model can accurately classify nearly all pixels on the test set, demonstrating strong performance on test data.

Test IoU- 0.8763.

The IoU score for the test set is slightly lower than those for the training and validation sets, suggesting the model is marginally less effective at identifying regions of interest in the test data. However, a score of 0.8763 is still high, indicating strong overall performance.

Test Dice Coefficient- 0.9064

The Dice coefficient for the test set is 0.9064, slightly lower than the training and validation scores. This suggests a marginal decrease in segmentation accuracy on completely unseen data, though the model's performance remains strong.

4.2.3 Predicted Images By PSPNet-Resnext50_32x4d and UNet-EfficientNetB7

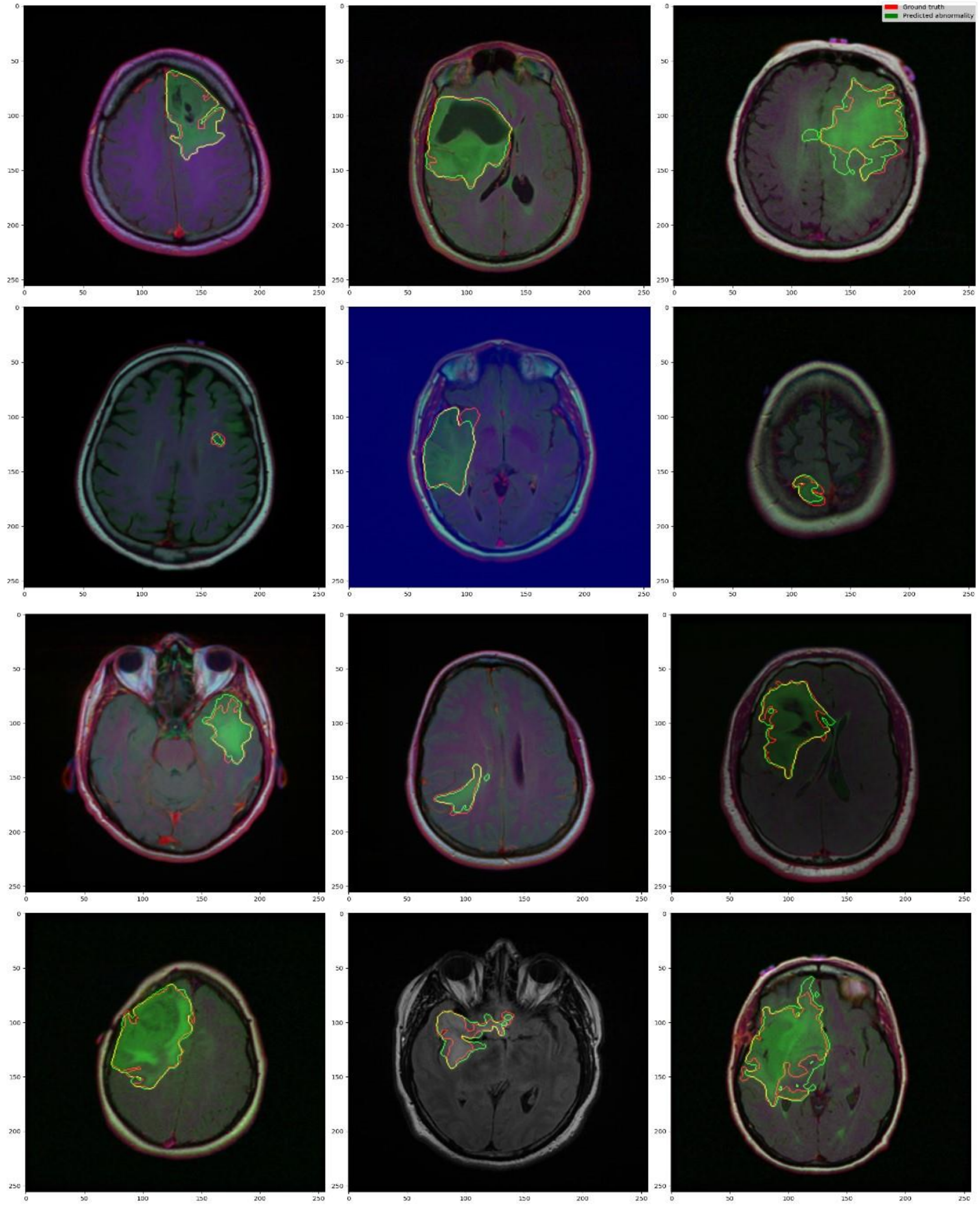


Figure 4.9: Predicted Images by UNet-EfficientNetB7

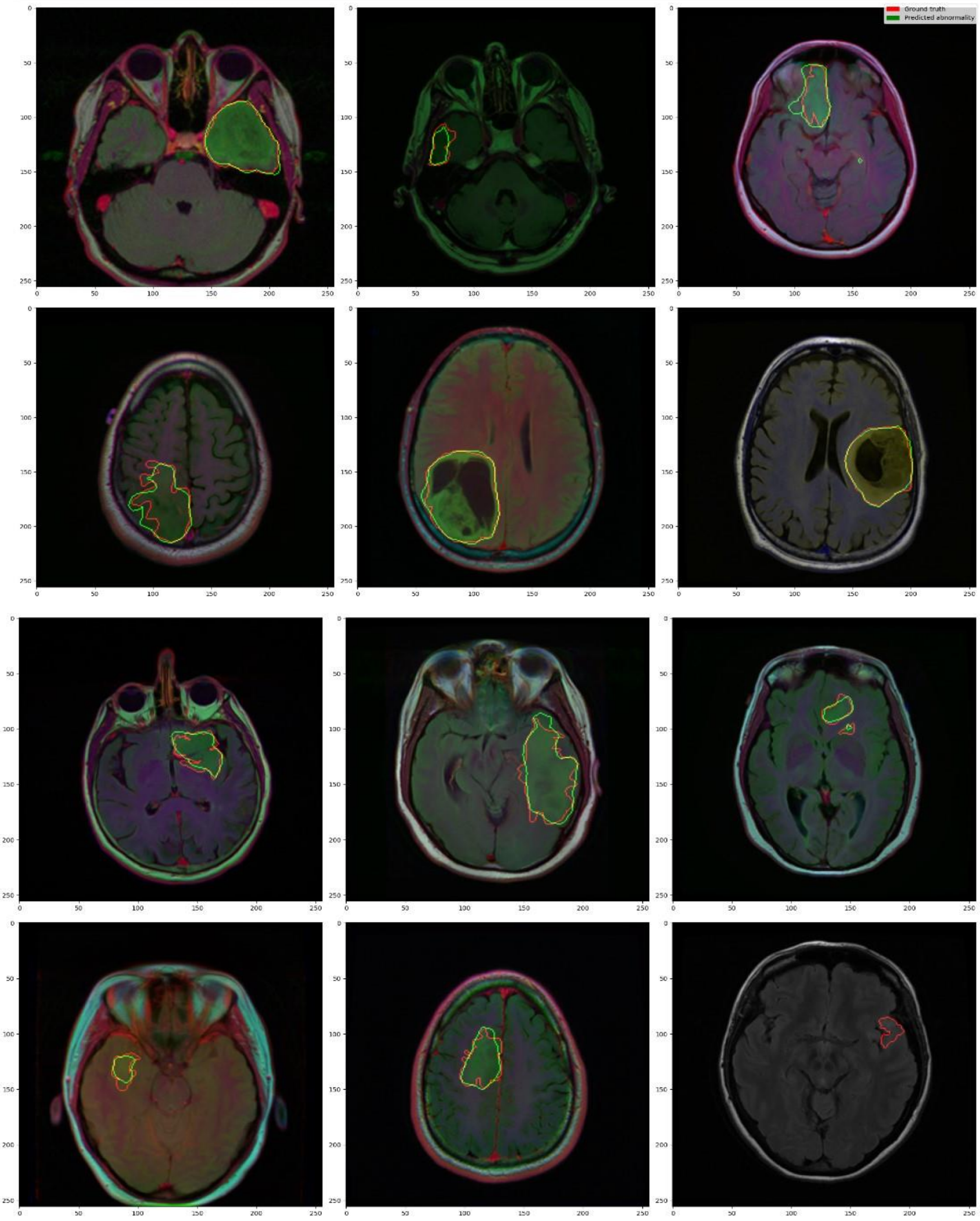


Figure 4.10: Predicted Images by PSPNet-Resnext50_32x4d

Here’s a comparative analysis between UNet-EfficientNetB7 and PSPNet-ResNeXt50_32x4d, both using transfer learning, based on Figures 4.9 and 4.10-

1. General Overview Of The Prediction -

Ground Truth (Red Contour)- The actual tumor areas as noted by medical specialists.

Predicted Tumor (Green Contour)- Regions predicted by UNet-EfficientNetB7 and the PSPNet-ResNeXt50_32x4d model. The overlap of the green and red outlines allows us to measure how well the model performs.

2. Prediction Accuracy –

UNet-EfficientNetB7-

The model accurately detects small, well-defined tumors, with predicted contours (green) closely matching the ground truth (red). For larger tumors, it shows mild under-segmentation, with predicted zones often smaller than the actual ground truth, particularly for complex tumor shapes.

PSPNet-ResNeXt50_32x4d-

PSPNet delivers high accuracy in detecting both small and large tumors, excelling in segmenting larger, irregular tumors. It outperforms UNet in border alignment and tumor coverage but may slightly over-segment intricate boundary regions.

3. Boundary Precision –

UNet-EfficientNetB7-

The model shows high boundary precision for smaller, simpler tumors, closely aligning with the ground truth. However, for larger or more complex tumors, it struggle with boundary precision, often under-segmenting and missing intricate tumor regions.

PSPNet-ResNeXt50_32x4d-

PSPNet has strong border precision for larger, irregularly shaped tumors, accurately capturing complex boundaries even in challenging cases. Over-segmentation is rare and has minimal effect on overall performance.

4. Handling Different Tumor Size and Shapes -

UNet-EfficientNetB7-

The model excels at segmenting tiny tumors, achieving near-perfect accuracy in detecting minor irregularities and well-defined shapes. However, UNet struggles with larger tumors, particularly at the edges, often failing to capture the entire area in more irregular or complex structures.

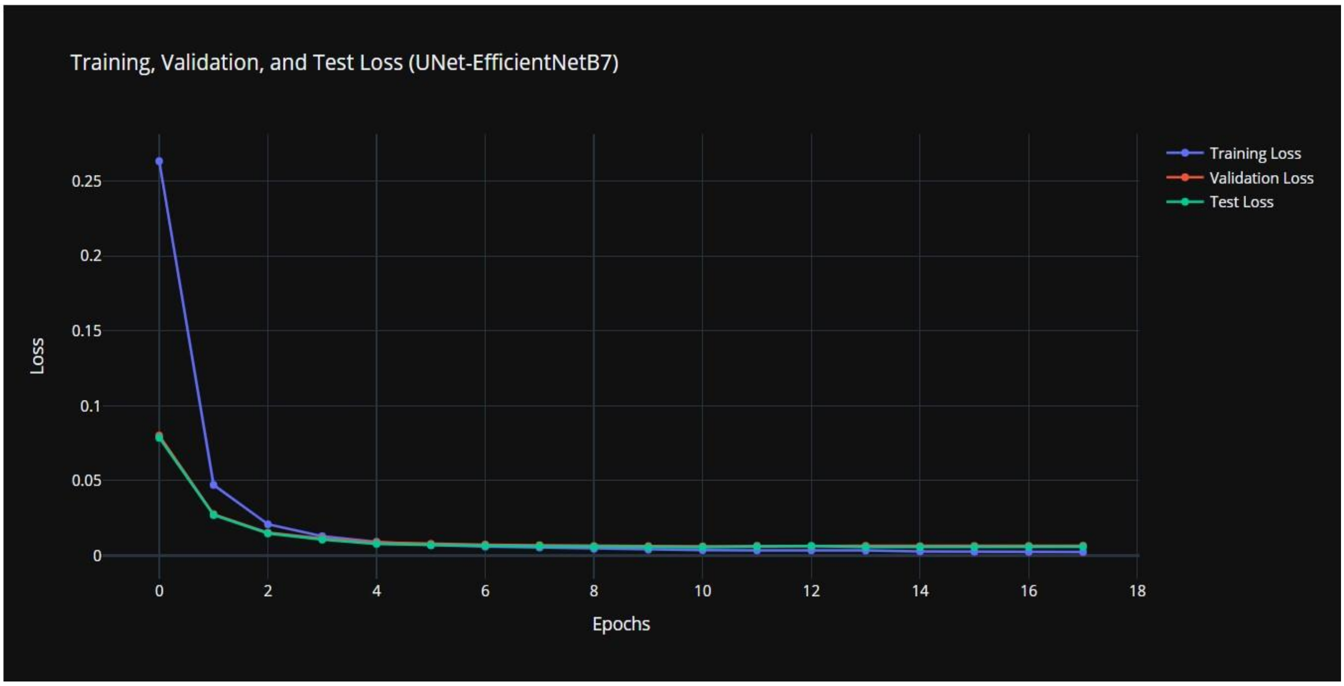
PSPNet-ResNeXt50_32x4d-

PSPNet is versatile, effectively handling both small and large tumors. It excels in segmenting larger, irregular tumors, reliably capturing the entire lesion with minimal boundary variations. The model demonstrates robustness, performing well even with highly irregular tumor shapes.

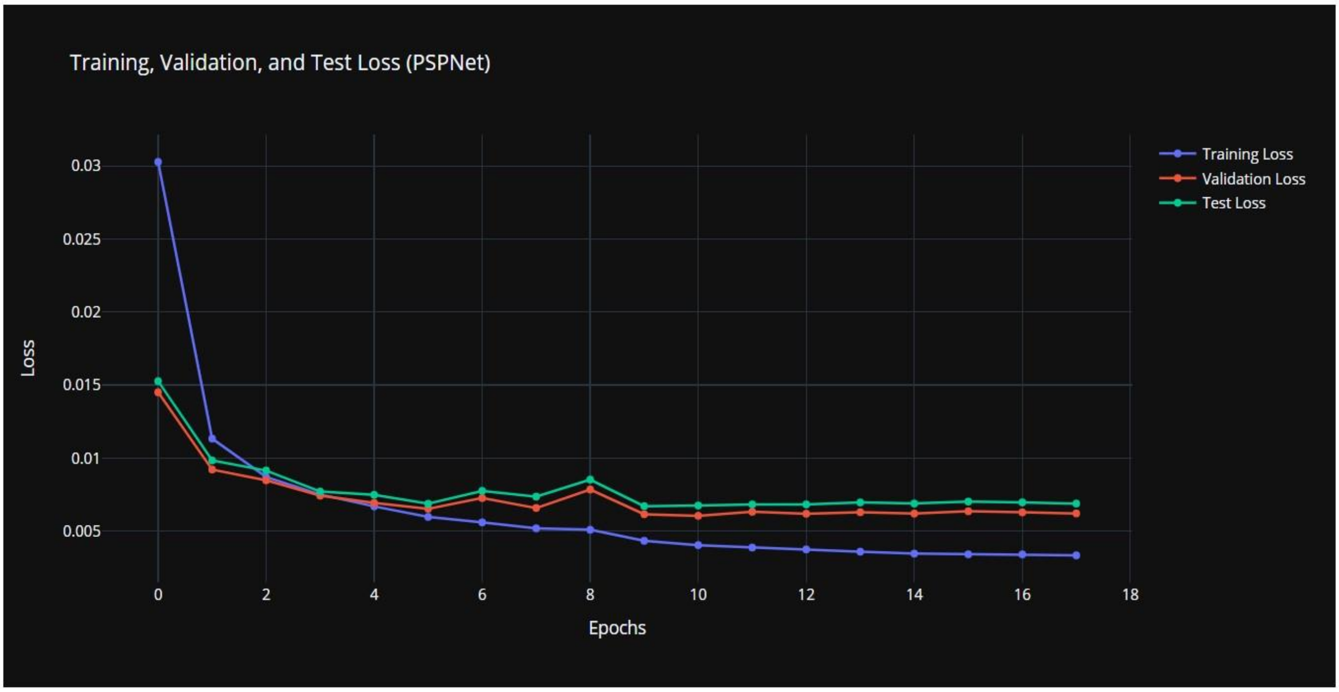
4.2.4 Visualization of Performance Metrics on Both PSPNet-Resnext50_32x4d and UNet-EfficientNetB7

4.2.4.1 Visualization On Loss on Train, Validation and Test Data

This graph shows the Loss across the Train, Test, and Validation datasets over 18 epochs for both UNet EfficientNetB7 and PSPNet-ResNeXt50_32x4d. Loss is crucial as it reflects the model's optimization process and provides a clear measure for evaluating and improving performance.



(a)



(b)

Figure 4.11: Epoch Vs Training, Validation and Test Loss (a) UNet EfficientNetB7 & (b) PSPNet-Resnext50_32x4d

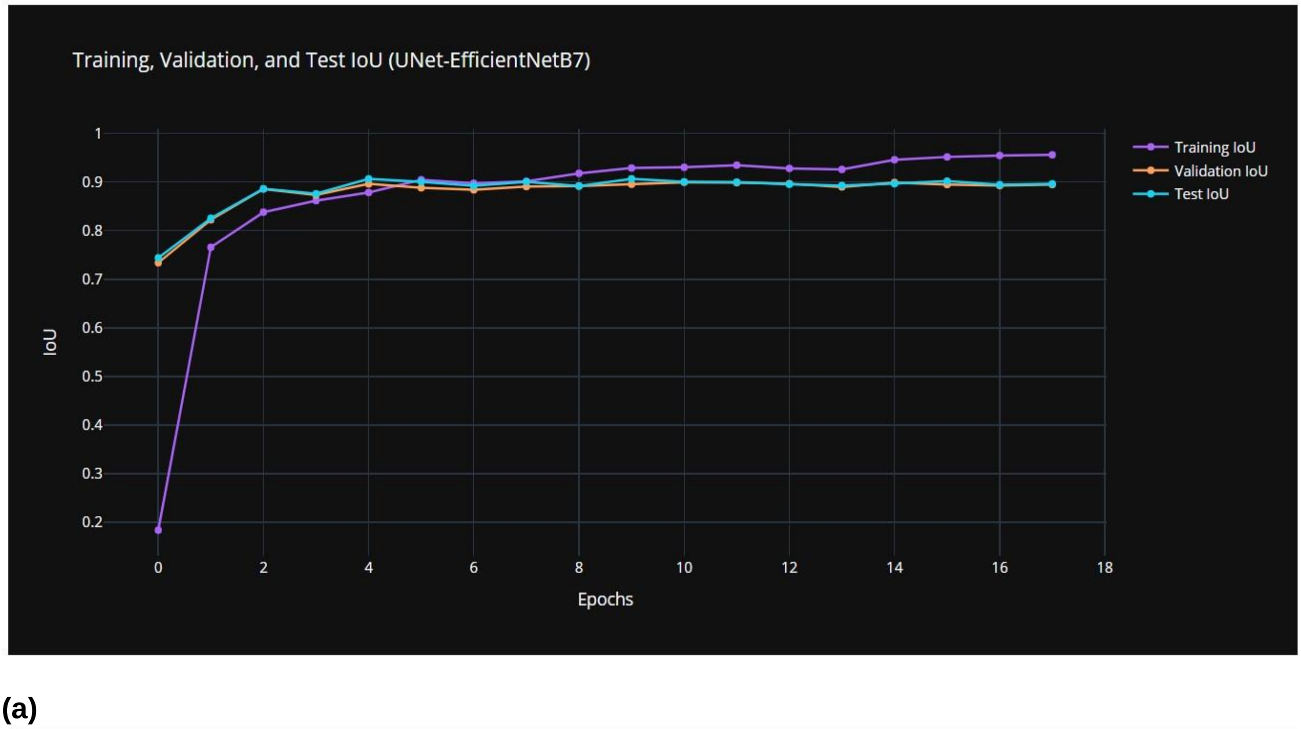
It can be seen in Figure 4.11 (a), The **UNet EfficientNetB7** model the training loss starts at high at around 0.28 and decreases sharply in the first few epochs, flattening out close to zero around epoch 6 and stabilizes around 0.01 in between epoch 16 and 18. Similar to Training loss both the Validation and Test loss follow the same pattern, a rapid decrease in the first few epochs and settles around 0.01 in between epoch 16 to epoch 18. The loss curves for training, validation, and test data stay close together throughout the training process, showing effective generalization with minimal overfitting.

In Figure 4.11 (b), for **PSPNet-Resnext50_32x4d** the training loss begins at 0.03 and starts decreasing in the first few epochs and begins to flatten around epoch 6, in between epochs 16 and 18, the training loss completely remains stable and low at the value of 0.005. The Validation and Test Loss starts around 0.015 and starts decreasing similarly to the Training loss. In Between epochs 6 and 8, both of the losses fluctuate slightly overall maintaining a steady value very close to 0.005.

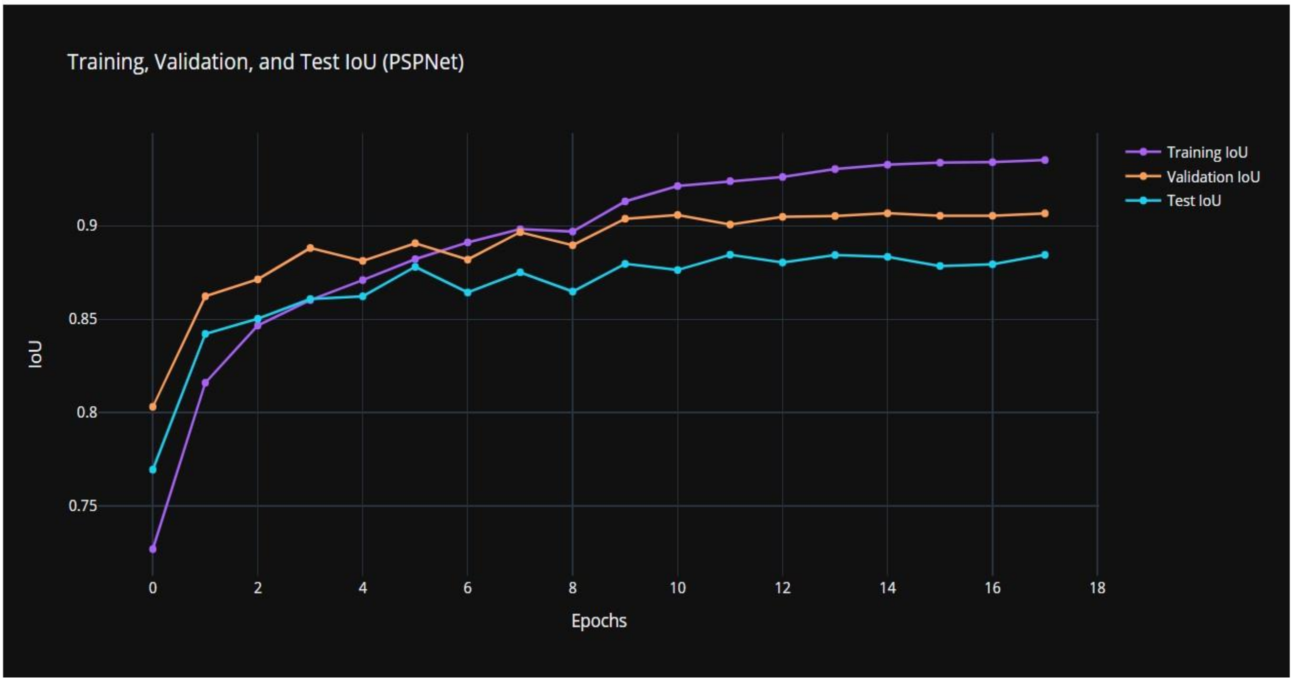
Overall, UNet-EfficientNetB7 produces smoother and more stable loss curves across training, validation, and test datasets. In contrast, PSPNet-ResNeXt50_32x4d shows more fluctuations in validation and test loss but achieves lower final loss values across all datasets. The fluctuations suggest minor instability and a slightly increased risk of overfitting.

4.2.4.2 Visualization On IoU (Intersection Over Union) on Train, Validation and Test Data

This plot in this section delineates about the **IoU** on Train, Test and validation dataset across the 18 epochs for both **UNet EfficientNetB7** and **PSPNet-ResNeXt50_32x4d**. IoU is important because It quantifies the overlap between the predicted bounding box or segmented region and the ground truth bounding box or annotated region from a dataset (encord, 2023).



(a)



(b)

Figure 4.12: Epoch Vs Training, Validation and Test IoU (a) UNet EfficientNetB7 & (b) Epoch Vs Training, Validation and Test IoU in PSPNet-Resnext50_32x4d

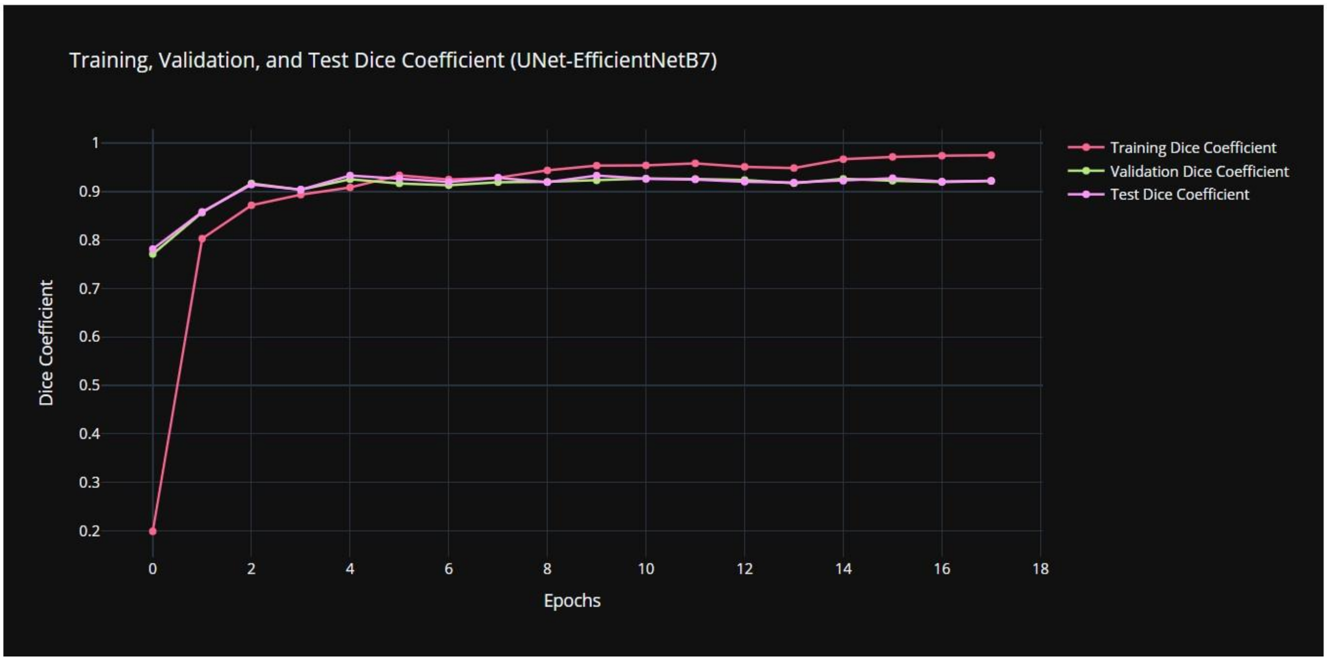
In the above figure, Figure 4.12 (a) For the **UNet EfficientNetB7** model the Training IOU starts at 0.2 but reaches at 0.85 with the first two epochs, epoch 0 and 2 which indicates that the model rapidly learns the spatial localization of the objects. The training IOU further improves from epoch 2 and stabilizes at around 0.92. After epoch 8 onward, the training IOU started increasing and reaches at 0.95 by epoch 18. In case of Validation and Test IOU both increases rapidly reaching at 0.9 within epoch 0 and 2. From epoch 2 to 8 the Validation and Test IoU is slightly below to Training IoU, stabilizing between 0.88 and 0.9, which indicates the model is still improving but with slower and there is a minimal overfitting within this region as the gap between training and validation and test is small. From epoch 8 to 18 both the Test and Validation stabilizes near 0.9. The consistency across the IoU values for training, validation, and test sets indicates that the model is generalizing well and is not overfitting.

For Figure 4.12 (b), In the case of the **PSPNet-ResNeXt50_32x4d** model the Training IoU jumps quickly from 0.25 to 0.85 within the first 2 epochs. From epoch 2 the Training IoU continues to improve and reaches at 0.9 at epoch 8. From epoch 8 it again increases and stabilizes at 0.95 by the end at epoch 18. The Validation and Test IoU start from 0.8 and 0.76 respectively both of them reach at around 0.85 and 0.80. From epoch 2 and 8, both of the Validation and Test IoU fluctuate and showing a slight dip in between epoch 4 and epoch 6 and stabilise between 0.9 and 0.87. This could indicate some overfitting, as the validation performance shows minor inconsistency. Both Validation and Test IoU stabilizes at 0.91 and 0.88. The model shows similar initial performance improvement but displays more fluctuations in the validation and test IoU during the training process, especially between epochs 4 and 8.

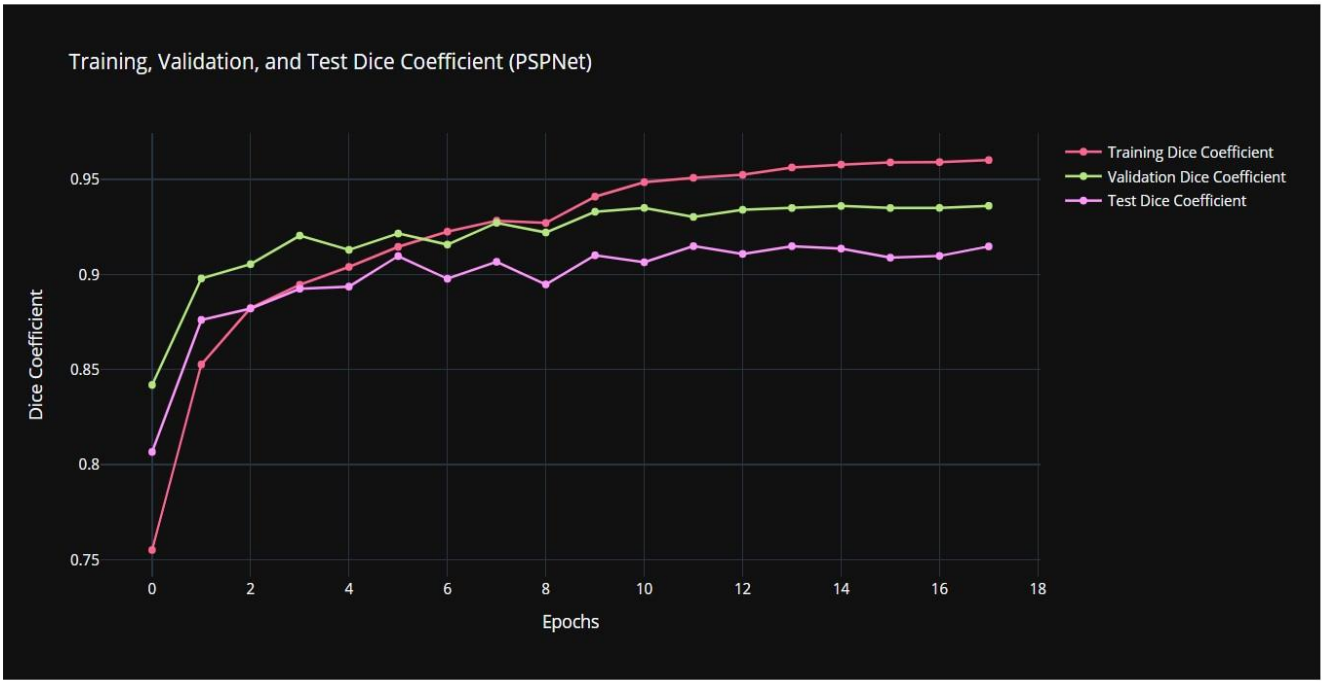
Overall, **UNet-EfficientNetB7** emerges as the winner in terms of IoU stability, and generalization. It performs more consistently across training, validation, and test sets, and shows less fluctuation and overfitting compared to **PSPNet-Resnext50_32x4d**.

4.2.4.3 Visualization On Dice Coefficient on Train, Validation and Test Data

The graphs in this section displays the Dice Coefficients for training, validation, and test data across 18 epochs for both UNet EfficientNetB7 and PSPNet-ResNeXt50_32x4d. The Dice Coefficient is an essential metric in segmentation tasks, quantifying the overlap between predicted and true regions.



(a)



(b)

Figure 4.13: Epoch Vs Training, Validation and Test Dice Coefficient in (a)UNet EfficientNetB7 & (b)PSPNet-Resnext50_32x4d

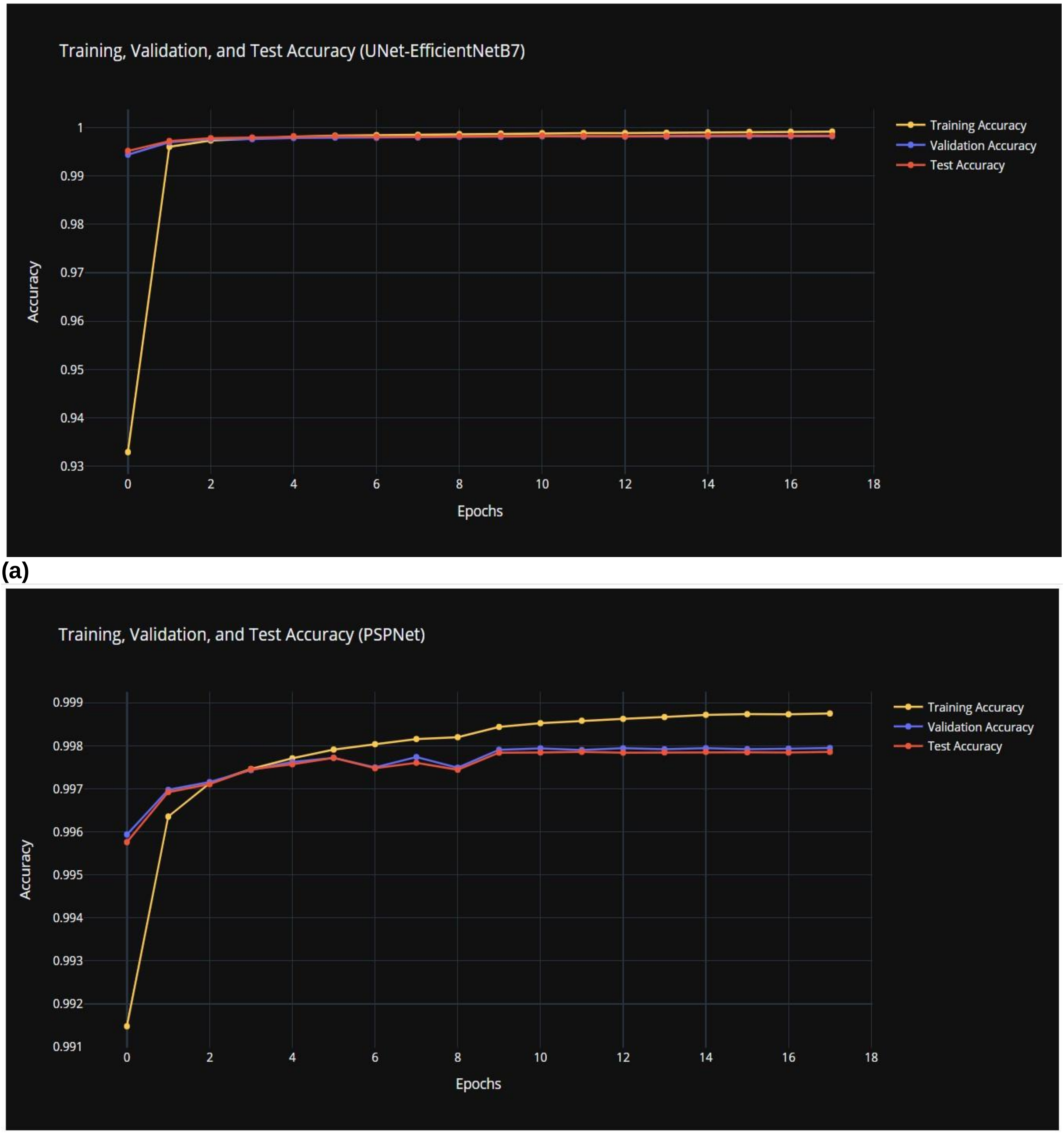
As can be seen in Figure 4.13 (a), the **UNet EfficientNetB7** model demonstrates a rapid increase in the training Dice Coefficient, surpassing 0.9 after just two epochs and stabilising at around 0.95 by the 10th epoch, indicating excellent fit to the training data. The validation and test Dice Coefficients closely follow this trend, stabilising around 0.93, suggesting strong generalisation and consistent performance across all datasets without significant overfitting. The near-identical performance on the training, validation, and test sets highlights the model's robustness and ability to maintain high prediction accuracy.

Similar to **UNet,** in Figure 4.13 (b) the training Dice Coefficient for **PSPNet** rises quickly in the initial epochs, exceeding 0.9 by the third epoch. However, it converges more gradually, peaking at around 0.95 by the 15th epoch, indicating that **PSPNet** may require more epochs to fully optimise. The validation and test Dice Coefficients exhibit more fluctuations compared to **UNet**, stabilising around 0.93 and 0.91. While the model generalises well, the fluctuations suggest minor instability across datasets and a slightly higher risk of overfitting.

Both models show a quick performance boost in early epochs. UNet EfficientNetB7 reaches higher and more stable Dice Coefficient values faster, stabilizing around 0.95 for training and 0.93 for validation and test datasets. PSPNet stabilizes at 0.95 for training but slightly lower at 0.91 for validation and test sets.

4.2.4.3 Visualization On Accuracy on Train, Validation and Test Data

In this section, we present the training, validation, and test accuracy. The graphs displayed below, visualised across 18 epochs, provide insights into how well each model generalises during training and evaluation phases. The accuracy curves are plotted to compare the models' ability to identify the tumour and background regions effectively, as well as how they handle unseen data (test set).



(a)

(b)

Figure 4.14: Epoch Vs Training, Validation and Test Accuracy in (a)UNet EfficientNetB7 & (b)PSPNet-Resnext50_32x4

We noticed that both **UNet EfficientNetB7** and **PSPNet-ResNeXt50_32x4d** show high performance in terms of training, validation, and test accuracy. However, there are subtle distinctions that highlight their different learning behaviors.

For both models, the **training accuracy** increases rapidly within the first few epochs (with training accuracy going from 0.93 to 0.99). The **UNet EfficientNetB7** model reaches an almost perfect training accuracy close to 1.0 by the second epoch and maintains this high performance throughout the training. Similarly, the **PSPNet-ResNeXt50_32x4d** model shows a steady rise in training accuracy, with a slight difference in how quickly it converges. The training accuracy increases from 0.991 to 0.998 by epoch 6, with a consistent improvement over time, indicating its capability to fit the training data well, though it slightly lags behind UNet in the initial stages.

Both models achieve consistently high **validation and test accuracy** after just a few epochs, closely mirroring their training performance. **UNet EfficientNetB7** reaches a validation accuracy that matches its training accuracy early on, maintaining stable and consistent performance across all epochs. **PSPNet-ResNeXt50_32x4d** also demonstrates high validation accuracy but with minor fluctuations around the eighth epoch, showing a slightly slower convergence. This suggests that UNet may generalise to unseen data more smoothly than PSPNet during certain points in the training process.

UNet EfficientNetB7 demonstrates better generalization, with nearly identical accuracy curves across training, validation, and test datasets from early epochs. This consistency suggests that UNet is less prone to overfitting compared to PSPNet, which experiences minor fluctuations in validation and test accuracy mid-training. While PSPNet-ResNeXt50_32x4d achieves near-perfect accuracy, it lacks the same level of seamless generalization as UNet.

4.2.4.4 Comparative analysis and Confusion Matrix for UNet EfficientNetB7 and PSPNet-ResNeXt50_32x4d models

In this section, a detailed comparison of **UNet EfficientNetB7** and **PSPNet-ResNeXt50_32x4d** is presented, using confusion matrices to evaluate their performance in segmenting tumour and background regions. Figure 4.15 illustrates the confusion matrices for both models, providing insights into their accuracy in classifying pixel data for background (Class 0) and tumour (Class 1) regions.

The confusion matrix summarizes correct and incorrect pixel classifications for each model. It highlights how well each model identifies background pixels and tumor regions. This analysis provides insights into their strengths and weaknesses in medical image segmentation, essential for creating reliable diagnostic tools. The comparison includes background prediction accuracy, tumor detection precision, and misclassification rates.

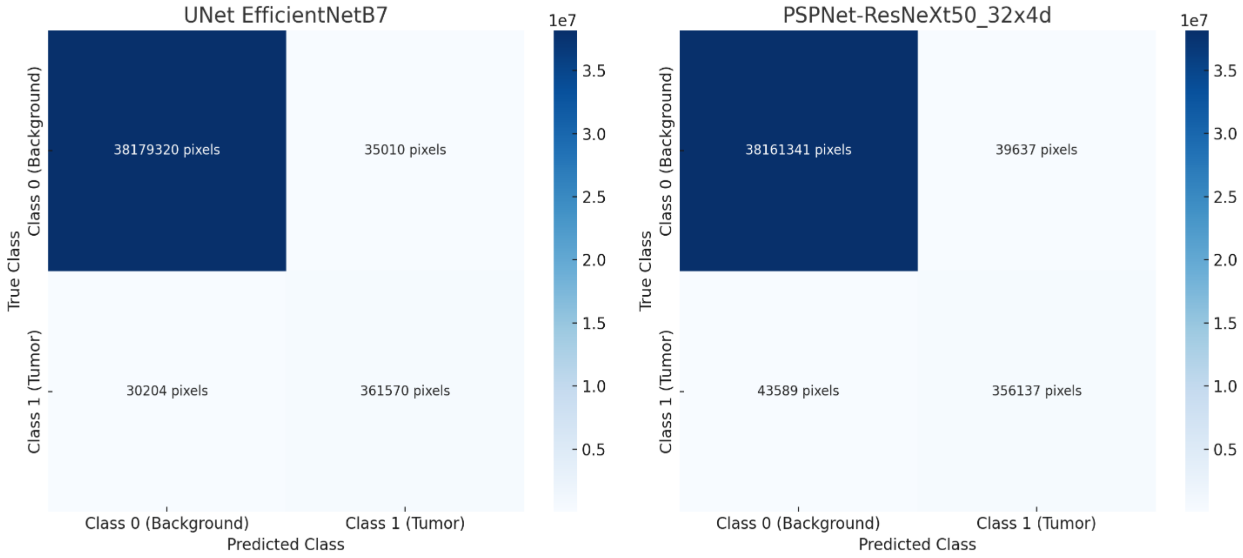


Figure 4.15: Confusion Matrix for UNet EfficientNetB7 and PSPNet-ResNeXt50_32x4d

Correct Background Prediction (Class 0):

Both models effectively predict background pixels, but UNet EfficientNetB7 has a slight edge over PSPNet-ResNeXt50_32x4d. UNet EfficientNetB7 correctly identifies 38,179,320 background pixels compared to PSPNet's 38,161,341. UNet also misclassifies fewer background pixels as tumors (35,010) compared to PSPNet (39,637), indicating that UNet is slightly better at avoiding false positives.

Correct Tumour Prediction (Class 1):

For tumor identification, both models are quite similar in performance. UNet EfficientNetB7 correctly classifies 361,570 tumor pixels, slightly outperforming PSPNet-ResNeXt50_32x4d, which predicts 356,137 tumor pixels. Although the difference is small, UNet EfficientNetB7 has a slight edge in accurately identifying tumor pixels, which is crucial for precision in medical imaging.

The misclassification results reveal a notable difference between the models. PSPNet-ResNeXt50_32x4d misclassifies 43,589 tumor pixels as background, whereas UNet EfficientNetB7 only misclassifies 30,204 tumor pixels as background. This indicates that UNet is better at detecting tumors and reducing the risk of missed diagnoses. Additionally, UNet EfficientNetB7 has fewer false positives, misclassifying 35,010 background pixels as tumors compared to PSPNet's 39,637. This means UNet's lower false positive rate minimizes unnecessary concerns and additional testing.

Table: Pixel wise comparison for correct and incorrect classifications.

		UNet_EfficientNet B7	PSPNet-ResNeXt50_32x4d
True Class	Predicted Class	Pixels	Pixels
Class 0 (Background)	Class 0 (Background): background is successfully predicted	38,179,320 pixels	38,161,341 pixels
	Class 1 (Tumor): background wrongly identified as tumours.	35,010 pixels	39,637 pixels
Class 1 (Tumour)	Class 0 (Background): tumour pixels were incorrectly identified as background	30,204 pixels	43,589 pixels
	Class 1 (Tumor): tumour pixels are properly identified as tumours	361,570 pixels	356,137 pixels

Overall, **UNet EfficientNetB7** shows a marginally better performance in both the accurate classification of tumour and background regions and in reducing misclassifications. **PSPNet-ResNeXt50_32x4d**, while close in performance, has a higher error rate in both false positives (misclassifying background as tumours) and false negatives (misclassifying tumours as background).

4.2.4.5 Classwise F1 Score UNet EfficientNetB7 and PSPNet-ResNeXt50_32x4d

Table and Figure 19 display the class-wise F1 scores for UNet EfficientNetB7 and PSPNet-ResNeXt50_32x4d, highlighting their performance in identifying background (Class 0) and tumors (Class 1). The F1 score, which balances precision and recall, provides a comprehensive measure of each model’s effectiveness in medical imaging, emphasizing the importance of accurately distinguishing both tumor and non-tumor regions.

Table: Classwise F1 Score for UNet EfficientNetB7 & PSPNet-Resnext50_32x4d

Model	Class 0 (background) F1 Score	Class 1 (tumour) F1 Score
UNet EfficientNetB7	0.999147	0.917278
PSPNet-ResNeXt50_32x4d	0.998107	0.895379

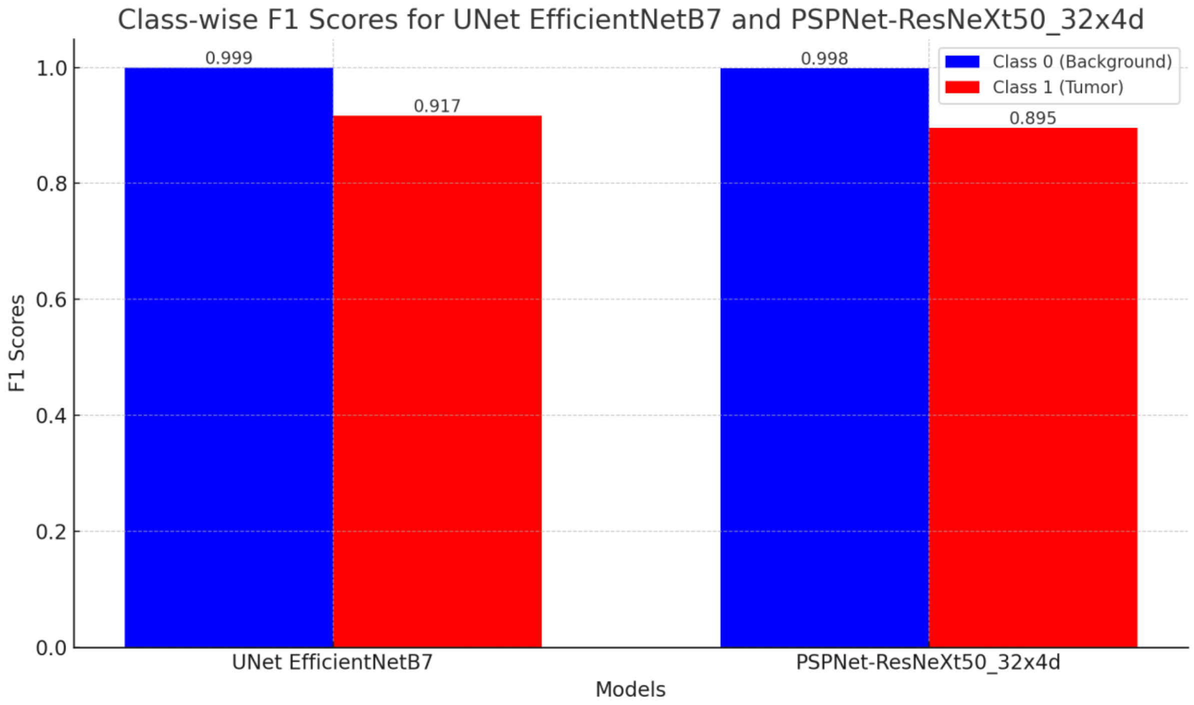


Figure 4.16: Classwise F1 Score for UnetEfficientB7 and PSPNet

UNet EfficientNetB7:

Class 0 (Background): UNet EfficientNetB7 achieves a high F1 score of 0.999, demonstrating excellent accuracy in identifying background pixels with minimal misclassifications.

Class 1 (Tumour): The F1 score for tumour classification is **0.917**, which, while still strong, is lower than the score for the background class. This suggests that while UNet EfficientNetB7 performs well in detecting tumour pixels, it faces more challenges compared to background classification. The lower score points to a slightly higher number of false positives or false negatives in tumour detection.

PSPNet-ResNeXt50_32x4d:

Class 0 (Background): PSPNet also achieves an impressive F1 score of **0.998** for the background class, demonstrating a strong ability to accurately classify non-tumour regions. This indicates that, similar to UNet, PSPNet is reliable when it comes to detecting background pixels.

Class 1 (Tumour): For tumour classification, PSPNet achieves an F1 score of **0.895**, slightly lower than UNet. This suggests that PSPNet is effective but slightly less precise in identifying tumour pixels, leading to more false positives or negatives compared to UNet EfficientNetB7. Although the difference is minimal, it could have implications in medical contexts, where accuracy in tumour detection is critical.

Both models excel at detecting background regions, with F1 scores near 1.0, indicating strong performance in identifying non-tumour areas. However, UNet EfficientNetB7 slightly outperforms PSPNet-ResNeXt50_32x4d in tumour classification, achieving an F1 score of 0.917 versus 0.895. This small advantage in tumour detection is crucial in medical imaging, as improved accuracy in identifying tumour regions can significantly impact diagnostic reliability. Thus, UNet EfficientNetB7's better performance in Class 1 (tumour) detection makes it potentially more suitable for precise tumour identification tasks.

4.2.4.6 Addressing The Research Questions-

In first chapter the research is driven by the following key questions:

1. How effective are deep learning models like PSPNet and UNet with EfficientNetB7 for segmenting brain tumors in MRI images in terms of accuracy, Dice score, and IoU?

2. Does transfer learning provide a significant advantage in brain MRI segmentation compared to models trained from scratch?

3. Which specific deep learning architecture—PSPNet or UNet EfficientNetB7—shows better generalisability and performance across training, validation, and test datasets?

While addressing the first question by comparing the performance of three models—PSPNet from scratch, PSPNet with ResNeXt50_32x4d, and UNet with EfficientNetB7—the transfer learning models outperform the PSPNet from scratch. Among them, UNet-EfficientNetB7 shows superior performance in IoU and Dice coefficients, excelling in generalization and segmentation accuracy compared to PSPNet with ResNeXt50_32x4d.

In case of second research question, The results show that the PSPNet model built from scratch struggles to identify the full extent of anomalies, leading to significant under-segmentation where the predicted mask is much smaller than the actual one. Its boundary precision is also poor, failing to capture the outlines of the abnormality and missing important tumor areas, a common issue in models that are not well-trained or optimized.

While in **UNet-EfficientNetB7** is highly effective at detecting and segmenting tumor areas, especially for minor and well-defined abnormalities, with high border precision. However, it shows slight under-segmentation for larger or more complex tumors, leading to some missed sections.

The **PSPNet with ResNeXt50_32x4d** is the most robust model for tumor segmentation, effectively handling both small and large tumors. It has the highest accuracy and boundary precision among the models, particularly excelling in capturing the contours of larger or irregular tumors.

Also, as mentioned before that both transfer learning models (UNet-EfficientNetB7 and PSPNet with ResNeXt50_32x4d) are less time-consuming and computationally expensive compared to the PSPNet model built from scratch.

In the case of the Third research question, The UNet with EfficientNetB7 outperforms both PSPNet with ResNeXt50_32x4d and PSPNet from scratch across training, validation, and test datasets. It produces smoother and more stable loss curves, with excellent IoU stability and generalization. The model shows consistent performance with minimal fluctuation and overfitting, achieving higher, stable Dice Coefficients and nearly identical accuracy curves across all datasets from early epochs.

4.3 Summary –

This chapter focuses on the performance evaluation metrics of the PSPNet from scratch, PSPNet Transfer Learning model, and UNet EfficientNetB7 model. It critically assesses the successes and failures of each model, addressing all research questions based on the findings.

Chapter 5: Discussion

In our discussion section we will discuss and evaluate the findings and do a comparative analysis in between the models. Then we will talk about the research gaps and how those gaps have been filled up using our models.

From the prediction of the brain tumor image segmentation task, we will be discussing every model.

5.1 Comparative analysis on prediction Brain MRI segmentation in different models

When handling different tumor sizes and shapes, the PSPNet model built from scratch struggles with both small and large tumors, particularly failing to capture the full shape of irregular or larger tumor areas.

UNet-EfficientNetB7 excels at segmenting small, simple-shaped tumors, but with larger or more irregular tumors, it struggles to fully capture the extent of the issue, though it remains effective overall.

PSPNet-ResNeXt50_32x4d is highly adaptable, handling both small and large tumors with ease. It outperforms PSPNet from scratch in segmenting large, irregular shapes, an area where UNet-EfficientNetB7 struggles.

5.2 Comparative Analysis on F1 score and Confusion matrix in between the transfer learning models-

The results show that both models had F1 scores near 1.0 for identifying background regions, demonstrating their reliability in distinguishing non-tumor areas. However, UNet EfficientNetB7 outperforms PSPNet-ResNeXt50_32x4d in tumor classification, with an F1 score of 0.917 compared to 0.895. Though the difference is small, it can significantly impact medical imaging by improving tumor detection and reducing the risk of missed diagnoses. Therefore, UNet is better suited for applications requiring precise tumor identification.

Overall, UNet EfficientNetB7 performs better in the confusion matrix by minimizing misclassifications and accurately distinguishing between background and tumor regions. While PSPNet with ResNeXt50_32x4d performs similarly, it has a higher error rate in false positives (misclassifying background as tumors) and false negatives (misclassifying tumors as background).

5.3 Addressing the Research Gaps-

Studies like Hafiz et al. (2023) have highlighted segmentation accuracy issues, particularly with smaller or less distinct tumor areas. In this research, the UNet with EfficientNetB7 model efficiently segmented small and simple-shaped tumors, while PSPNet with ResNeXt50_32x4d performed well on both small and large tumor areas.

Ramasamy, Singh, and Yuan (2023) and Osman Özkaraca et al. (2023) proposed models that are computationally expensive and time-consuming due to their depth and complexity. In contrast, our models—PSPNet with ResNeXt50_32x4d and UNet with EfficientNetB7—are less computationally costly and faster, as they utilize transfer learning and are pre-trained on the ImageNet dataset, as discussed in the methodology section.

While some studies report excellent accuracy, they lack a thorough evaluation using key metrics like F1-score, sensitivity, specificity, recall, and precision. For instance, Saeedi et al. (2023) and Marco Antonio Gómez-Guzmán et al. (2023) did not thoroughly assess their models with these metrics, limiting their practical value in medical diagnostics. In contrast, we evaluated these metrics when deploying transfer learning methods to ensure a more comprehensive assessment.

5.4 Summary -

The discussion chapter provides a concise overview of the findings collected from three models: PSPNet from Scratch, PSPNet-ResNeXt50_32x4d, and UNet EfficientB7. This chapter also addressed the research gaps which had been discussed before.

Chapter 6: Conclusion

This chapter provides an overview of the dissertation, highlights the research contributions made, and explores potential future research and development stemming from the dissertation.

6.1 Summary of The Dissertation –

The brain is a vital organ, and brain tumors are a dangerous form of cancer. Medical imaging, such as MRI, plays a key role in diagnosis, treatment planning, and monitoring. Tumor segmentation is challenging due to variations in shape, size, location, and appearance, especially with gliomas, classified as High-Grade Glioma (HGG) and Low-Grade Glioma (LGG). This study builds on previous research that includes both HGG and LGG segmentation.

In Chapter 1, the literature review highlights various approaches like machine learning, deep learning, CNN-based techniques, transfer learning, and data augmentation. Research gaps identified in the review are addressed later in the discussion section. Chapter Methods covers methodology, where three advanced models—PSPNet from scratch, PSPNet with transfer learning, and UNet EfficientNetB7—were used for image segmentation. It also discusses dataset collection, data pre-processing, and model application. The Result evaluates the performance, success, and failure of the models, while the Discussion chapter summarizes the findings and provides a comparative analysis based on the models' performance.

6.2 Research Contributions –

The improved models have several significant contributions in the research field. First we will discuss about the academic field and then we will discuss about the practical field.

1. Academic Field -

The developed models contribute significantly to ongoing academic research on brain tumor segmentation by utilizing advanced architectures such as UNet with EfficientNetB7 and PSPNet with ResNeXt50_32x4d. This exploration of sophisticated architectures pushes the boundaries of medical image segmentation and may inspire further research into transfer learning and hybrid models in medical applications. The integration of EfficientNetB7 with UNet offers a novel approach, combining EfficientNet's feature extraction efficiency with UNet's segmentation capabilities. Similarly, building PSPNet from scratch and using ResNeXt50_32x4d provides valuable insights into how different architectures handle medical image segmentation, focusing on accuracy, computational efficiency, and generalization across datasets.

The study also establishes specific evaluation criteria—such as IoU, Dice Coefficient, Binary Cross-Entropy, and Loss Curves—setting a benchmark for future researchers. By providing baseline data and addressing performance bottlenecks, this work will serve as a valuable reference for those developing similar models in the field.

2. Practical Field –

In real-world clinical settings, UNet-EfficientNetB7 is particularly valuable for precision segmentation of small, well-defined tumors, aiding in early detection, which is critical for timely intervention. Its efficiency in handling smaller tumors makes it an excellent tool for clinical use. PSPNet-ResNeXt50_32x4d, on the other hand, excels at segmenting larger, more complex tumors, providing better coverage and boundary identification, making it ideal for monitoring tumor growth or spread.

While these models are designed for brain tumor segmentation, the transfer learning approaches—using EfficientNet and PSPNet—are adaptable to various medical imaging tasks, such as detecting liver lesions, lung nodules, or cardiac anomalies, making the work applicable across different medical fields.

For resource-limited settings, UNet-EfficientNetB7 offers a cost-efficient solution due to its high accuracy and efficiency. It could help hospitals with limited access to radiologists by providing an effective tool for initial screening and diagnosis, thus democratizing access to advanced diagnostic technologies.

6.3 Future research and development

Future research could focus on fine-tuning the hyperparameters of UNet-EfficientNetB7 and PSPNet-ResNeXt50_32x4d to improve their accuracy, particularly in identifying tumor boundaries. This might involve testing with larger, more diverse datasets to address issues like over- or under-segmentation.

To enhance generalization, future studies could train and test the models on a broader range of brain tumor MRI datasets, incorporating various tumor types, imaging modalities, and scanner characteristics (e.g., CT or PET scans). This would improve the models' adaptability to different clinical scenarios.

Expanding transfer learning techniques could also be a promising avenue. Preparing the models for more general medical imaging tasks could help them develop more robust features, improving their ability to detect complex or subtle tumors.

Addressing class imbalance is another critical area for future study. Advanced data augmentation techniques, especially for underrepresented classes like smaller or rarer tumors, could boost the models' robustness across different tumor types and imaging conditions.

While PSPNet from scratch contributes to the segmentation process, it has notable drawbacks, including lower segmentation accuracy compared to UNet-EfficientNetB7 and PSPNet-ResNeXt50_32x4d, and its computational expense—it took 4 to 5 hours to train in this study. Future research could speed up training by using data and model parallelism to distribute workloads across multiple GPUs or workstations. More advanced data augmentation could also reduce the need for extensive training, improving both robustness and segmentation accuracy.

Combining PSPNet with other architectures like UNet or ResNet might also enhance performance. By integrating PSPNet's Spatial Pyramid Pooling with UNet's efficient encoding-decoding process, future models could achieve better segmentation with fewer computational resources.

6.4 Personal Reflections

This dissertation on Brain MRI Segmentation using advanced deep-learning models has significantly expanded my knowledge of segmentation algorithms such as PSPNet, UNet with EfficientNetB7, and the transfer learning process. It also enhanced my problem-solving skills in handling advanced deep-learning projects.

One of my key strengths discovered during this work is the ability to work effectively under pressure and maintain patience while training deep learning models and fine-tuning hyperparameters for optimal results. I experimented with adjusting batch sizes and epochs for each model to achieve the best performance possible.

Additionally, I identified areas for improvement, particularly in my understanding of frameworks. I have experience with both TensorFlow and PyTorch, but I noticed that most researchers in the field prefer PyTorch due to its agility, efficiency, and superior dynamic visualization compared to TensorFlow. Moving forward, I aim to focus more on PyTorch in future deep-learning projects to gain deeper expertise in this framework.

References

Ahmed Fawzy Gad (2020). *Accuracy, Precision, and Recall in Deep Learning | Paperspace Blog*. [online] Paperspace Blog. Available at: https://blog.paperspace.com/deep-learning-metrics-precision-recall-accuracy/ [Accessed 3 Sep. 2024].

Aloraini, M., Khan, A., Suliman Aladhadh, Habib, S., Alsharekh, M.F. and Islam, M. (2023). Combining the Transformer and Convolution for Effective Brain Tumor Classification Using MRI Images. *Applied Sciences*, [online] 13(6), pp.3680–3680. doi:https://doi.org/10.3390/app13063680.

Amjad Rehman Khan, Khan, S., Majid Harouni, Abbasi, R., Iqbal, S. and Mehmood, Z. (2021). Brain tumor segmentation using K-means clustering and deep learning with synthetic data augmentation for classification. *Microscopy Research and Technique*, [online] 84(7), pp.1389–1399. doi:https://doi.org/10.1002/jemt.23694.

Anand, V., Gupta, S., Gupta, D., Yonis Gulzar, Xin, Q., Juneja, S., Shah, A. and Shaikh, A. (2023). Weighted Average Ensemble Deep Learning Model for Stratification of Brain Tumor in MRI Images. *Diagnostics*, [online] 13(7), pp.1320–1320. doi:https://doi.org/10.3390/diagnostics13071320.

Anu Singha and Venkateswaran, V. (2023). *A Study on Effective Segmentation Network in MRI Images for Diagnosis of Brain Tumor*. [online] ResearchGate. Available at: https://www.researchgate.net/publication/379130167_A_Study_on_Effective_Segmentation_Network_in_MRI_Images_for_Diagnosis_of_Brain_Tumor [Accessed 1 Sep. 2024].

Aswani, K. and D. Menaka (2021). A dual autoencoder and singular value decomposition based feature optimization for the segmentation of brain tumor from MRI images. *BMC Medical Imaging*, [online] 21(1). doi:https://doi.org/10.1186/s12880-021-00614-3.

Bairagi, V.K., Pratima Purushottam Gumaste, Rajput, S.H. and None Chethan K. S (2023). Automatic brain tumor detection using CNN transfer learning approach. *Medical & Biological Engineering & Computing*, [online] 61(7), pp.1821–1836. doi:https://doi.org/10.1007/s11517-023-02820-3.

Buda, M. (2019). *Brain MRI segmentation*. [online] Kaggle.com. Available at: https://www.kaggle.com/datasets/mateuszbuda/lgg-mri-segmentation [Accessed 2 Sep. 2024].

Chen, C., Liu, X., Ding, M., Zheng, J. and Li, J. (2019). 3D Dilated Multi-fiber Network for Real-Time Brain Tumor Segmentation in MRI. *Lecture notes in computer science*, [online] pp.184–192. doi:https://doi.org/10.1007/978-3-030-32248-9_21.

Chen, S., Ding, C. and Liu, M. (2019). Dual-force convolutional neural networks for accurate brain tumor segmentation. *Pattern Recognition*, [online] 88, pp.90–100. doi:https://doi.org/10.1016/j.patcog.2018.11.009.

Developer, esri (2016). *How PSPNet works? | ArcGIS API for Python*. [online] Arcgis.com. Available at: https://developers.arcgis.com/python/guide/how-pspnet-works/ [Accessed 2 Sep. 2024].

Dufumier, B., Gori, P., Battaglia, I., Victor, J., Grigis, A. and Duchesnay, E. (2021). *Benchmarking CNN on 3D Anatomical Brain MRI: Architectures, Data Augmentation and Deep Ensemble Learning*. [online] arXiv.org. Available at: https://arxiv.org/abs/2106.01132 [Accessed 1 Sep. 2024].

encord (2023). *What is Intersection over Union (IoU)? | Definition | Encord | Encord*. [online] Encord.com. Available at: https://encord.com/glossary/iou-definition/#:~:text=IOU%20provides%20a%20measure%20of,of%20algorithms%20for%20improved%20results.&text=A%20higher%20IOU%20value%20indicates,reflecting%20a%20more%20accurate%20model. [Accessed 6 Sep. 2024].

Guan, X., Yang, G., Ye, J., Yang, W., Xu, X., Jiang, W. and Lai, X. (2022). 3D AGSE-VNet: an automatic brain tumor MRI data segmentation framework. *BMC Medical Imaging*, [online] 22(1). doi:https://doi.org/10.1186/s12880-021-00728-8.

Hafiz, Masood, T., Jaffar, A., Rashid, M. and Akram, S. (2023). Improved Multiclass Brain Tumor Detection via Customized Pretrained EfficientNetB7 Model. *IEEE Access*, [online] 11, pp.117210–117230. doi:https://doi.org/10.1109/access.2023.3325883.

Hao, R., Namdar, K., Liu, L. and Farzad Khalvati (2021). A Transfer Learning–Based Active Learning Framework for Brain Tumor Classification. *Frontiers in Artificial Intelligence*, [online] 4. doi:https://doi.org/10.3389/frai.2021.635766.

Hu, X., Luo, W., Hu, J., Guo, S., Huang, W., Scott, M.R., Wiest, R., Dahlweid, M. and Reyes, M. (2020). Brain SegNet: 3D local refinement network for brain lesion segmentation. *BMC Medical Imaging*, [online] 20(1). doi:https://doi.org/10.1186/s12880-020-0409-2.

Huynh, N. (2023). *Understanding Evaluation Metrics in Medical Image Segmentation*. [online]

Medium. Available at: https://medium.com/@nghihuynh_37300/understanding-evaluation-metrics-in-medical-image-segmentation-d289a373a3f#:~:text=The%20Intersection%2Dover%2DUnion%20(,scientific%20publictions%20for%20MIS%20evaluation [Accessed 3 Sep. 2024].

Isensee, F., Jaeger, P.F., Full, P.M., Vollmuth, P. and Maier-Hein, Klaus H (2020). *nnU-Net for Brain Tumor Segmentation*. [online] arXiv.org. Available at: https://arxiv.org/abs/2011.00848 [Accessed 1 Sep. 2024].

Jiang, M., Zhai, F. and Kong, J. (2021). A novel deep learning model DDU-net using edge features to enhance brain tumor segmentation on MR images. *Artificial Intelligence in Medicine*, [online] 121, pp.102180–102180. doi:https://doi.org/10.1016/j.artmed.2021.102180.

Khaled Bousabarah, Ruge, M., Brand, J.-S., Mauritius Hoevels, Rueß, D., Borggrefe, J., Nils Große Hokamp, Veerle Visser-Vandewalle, Maintz, D., Harald Treuer and Kocher, M. (2020). Deep convolutional neural networks for automated segmentation of brain metastases trained on clinical data. *Radiation Oncology*, [online] 15(1). doi:https://doi.org/10.1186/s13014-020-01514-6.

Kong, D., Liu, X., Wang, Y., Li, D. and Xue, J. (2022). 3D hierarchical dual-attention fully convolutional networks with hybrid losses for diverse glioma segmentation. *Knowledge-Based Systems*, [online] 237, pp.107692–107692. doi:https://doi.org/10.1016/j.knosys.2021.107692.

Kundu, R. (2024). *F1 Score in Machine Learning: Intro & Calculation*. [online] V7labs.com. Available at: https://www.v7labs.com/blog/f1-score-guide [Accessed 3 Sep. 2024].

Li, M., Kuang, L., Xu, S. and Sha, Z. (2019). *Brain Tumor Detection Based on Multimodal Information Fusion and Convolutional Neural Network*. [online] ResearchGate. Available at: https://www.researchgate.net/publication/338132223_Brain_Tumor_Detection_Based_on_Multimodal_Information_Fusion_and_Convolutional_Neural_Network [Accessed 1 Sep. 2024].

Liu, Y., Du, J., Vong, C.-M., Yue, G., Yu, J., Wang, Y., Lei, B. and Wang, T. (2022). Scale-adaptive super-feature based MetricUNet for brain tumor segmentation. *Biomedical Signal Processing and Control*, [online] 73, pp.103442–103442. doi:https://doi.org/10.1016/j.bspc.2021.103442.

Louis, D.N., Perry, A., Reifenberger, G., Andreas von Deimling, Figarella-Branger, D., Cavenee, W.K., Ohgaki, H., Wiestler, O.D., Kleihues, P. and Ellison, D.W. (2016). The 2016 World Health Organization Classification of Tumors of the Central Nervous System: a summary.

Acta Neuropathologica, [online] 131(6), pp.803–820. doi:https://doi.org/10.1007/s00401-016-1545-1.

Ma, Z., Zhou, S., Wu, X., Zhang, H., Yan, W., Sun, S. and Zhou, J. (2019). Nasopharyngeal carcinoma segmentation based on enhanced convolutional neural networks using multi-modal metric learning. *Physics in Medicine and Biology*, [online] 64(2), pp.025005–025005. doi:https://doi.org/10.1088/1361-6560/aaf5da.

Marco Antonio Gómez-Guzmán, Jiménez-Beristaín, L., Enrique Efren García-Guerrero, Oscar Roberto López-Bonilla, Ulises Jesús Tamayo-Perez, José Jaime Esqueda-Elizondo, Palomino-Vizcaino, K. and Everardo Inzunza-González (2023). Classifying Brain Tumors on Magnetic Resonance Imaging by Using Convolutional Neural Networks. *Electronics*, [online] 12(4), pp.955–955. doi:https://doi.org/10.3390/electronics12040955.

Md. Alamin Talukder, Md. Manowarul Islam, Md. Ashraf Uddin, Akhter, A., Alamgir, M., Aryal, S., Abdulllah, A., Khondokar Fida Hasan and Mohammad Ali Moni (2023). An efficient deep learning model to categorize brain tumor using reconstruction and fine-tuning. *Expert Systems with Applications*, [online] 230, pp.120534–120534. doi:https://doi.org/10.1016/j.eswa.2023.120534.

Menze, B.H., Jakab, A., Bauer, S., Jayashree Kalpathy-Cramer, Farahani, K., Kirby, J., Yuliya Burren, Porz, N., Johannes Slotboom, Wiest, R., Levente Lanczi, Gerstner, E., Weber, M.-A., Tal Arbel, Avants, B.B., Ayache, N., Buendia, P., D. Louis Collins, Cordier, N. and Corso, J.J. (2015). The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS). *IEEE Transactions on Medical Imaging*, [online] 34(10), pp.1993–2024. doi:https://doi.org/10.1109/tmi.2014.2377694.

Mohammad Ashraf Ottom, Hanif Abdul Rahman and Dinov, I.D. (2022). Znet: Deep Learning Approach for 2D MRI Brain Tumor Segmentation. *IEEE Journal of Translational Engineering in Health and Medicine*, [online] 10, pp.1–8. doi:https://doi.org/10.1109/jtehm.2022.3176737.

Muhammad Junaid Ali, Raza, B. and Ahmad Raza Shahid (2021). Multi-level Kronecker Convolutional Neural Network (ML-KCNN) for Glioma Segmentation from Multi-modal MRI Volumetric Data. *Journal of Digital Imaging*, [online] 34(4), pp.905–921. doi:https://doi.org/10.1007/s10278-021-00486-7.

N. Varuna Shree and Kumar, R. (2018). Identification and classification of brain tumor MRI images with feature extraction using DWT and probabilistic neural network. *Brain Informatics*, [online] 5(1), pp.23–30. doi:https://doi.org/10.1007/s40708-017-0075-5.

Neubauer, A., Hongwei Bran Li, Wendt, J., Schmitz-Koep, B., Menegaux, A., Schinz, D., Menze, B., Zimmer, C., Sorg, C. and Hedderich, D.M. (2022). Efficient Claustrum Segmentation in T2-weighted Neonatal Brain MRI Using Transfer Learning from Adult Scans. *Clinical Neuroradiology*, [online] 32(3), pp.665–676. doi:https://doi.org/10.1007/s00062-021-01137-8.

Nilesh Bhaskarrao Bahadure, Arun Kumar Ray and Har Pal Thethi (2017). Image Analysis for MRI Based Brain Tumor Detection and Feature Extraction Using Biologically Inspired BWT and SVM. *International Journal of Biomedical Imaging*, [online] 2017, pp.1–12. doi:https://doi.org/10.1155/2017/9749108.

Osman Özkaraca, Okan İhsan Bağrıaçık, Hüseyin Gürüler, Khan, F., Hussain, J., Khan, J. and Laila, U. e (2023). Multiple Brain Tumor Classification with Dense CNN Architecture Using Brain MRI Images. *Life*, [online] 13(2), pp.349–349. doi:https://doi.org/10.3390/life13020349.

Prince Priya Malla, Sahu, S. and Alutaibi, A.I. (2023). Classification of Tumor in Brain MR Images Using Deep Convolutional Neural Network and Global Average Pooling. *Processes*, [online] 11(3), pp.679–679. doi:https://doi.org/10.3390/pr11030679.

Ramasamy, G., Singh, T. and Yuan, X. (2023). *Multi-Modal Semantic Segmentation Model using Encoder Based Link-Net Architecture for BraTS 2020 Challenge*. [online] ResearchGate. Available at: https://www.researchgate.net/publication/367606619_Multi-Modal_Semantic_Segmentation_Model_using_Encoder_Based_Link-Net_Architecture_for_BraTS_2020_Challenge [Accessed 2 Sep. 2024].

Rasheed, Z., Ma, Y.-K., Ullah, I., Tamara Al Shloul, Ahsan Bin Tufail, Yazeed Yasin Ghadi, Muhammad Zubair Khan and Mohamed, H.G. (2023). Automated Classification of Brain Tumors from Magnetic Resonance Imaging Using Deep Learning. *Brain Sciences*, [online] 13(4), pp.602–602. doi:https://doi.org/10.3390/brainsci13040602.

Richmond Alake (2023). *Loss Functions in Machine Learning Explained*. [online] Datacamp.com. Available at: https://www.datacamp.com/tutorial/loss-function-in-machine-learning?dc_referrer=https%3A%2F%2Fwww.google.com%2F [Accessed 3 Sep. 2024].

Ronneberger, O., Fischer, P. and Brox, T. (2015). *U-Net: Convolutional Networks for Biomedical Image Segmentation*. [online] arXiv.org. Available at: https://arxiv.org/abs/1505.04597 [Accessed 3 Sep. 2024].

S. Deepak and Ameer, P.M. (2019). Brain tumor classification using deep CNN features via transfer learning. *Computers in Biology and Medicine*, [online] 111, pp.103345–103345. doi:https://doi.org/10.1016/j.compbiomed.2019.103345.

Saeedi, S., Sorayya Rezayi, Keshavarz, H. and Sharareh R. Niakan Kalhori (2023). MRI-based brain tumor detection using convolutional deep learning methods and chosen machine learning techniques. *BMC Medical Informatics and Decision Making*, [online] 23(1). doi:https://doi.org/10.1186/s12911-023-02114-6.

Sajid, S., Hussain, S. and Sarwar, A. (2019). Brain Tumor Detection and Segmentation in MR Images Using Deep Learning. *Arabian Journal for Science and Engineering*, [online] 44(11), pp.9249–9261. doi:https://doi.org/10.1007/s13369-019-03967-8.

Sajjad, M., Khan, S., Muhammad, K., Wu, W., Ullah, A. and Sung Wook Baik (2019). Multi-grade brain tumor classification using deep CNN with extensive data augmentation. *Journal of Computational Science*, [online] 30, pp.174–182. doi:https://doi.org/10.1016/j.jocs.2018.12.003.

Schumacher, D. (2023). *PSPNet*. [online] SERP AI. Available at: https://serp.ai/pspnet/ [Accessed 2 Sep. 2024].

Sharma, N. (2024). *Understanding and Applying F1 Score: AI Evaluation Essentials with Hands-On Coding Example*. [online] Arize AI. Available at: https://arize.com/blog-course/f1-score/#:~:text=F1%20score%20is%20a%20measure,better%20understanding%20of%20model%20performance. [Accessed 3 Sep. 2024].

Sinkus, R., Adomas Bunevicius, Schregel, K., Golby, A. and Patz, S. (2020). REVIEW: MR elastography of brain tumors. *NeuroImage Clinical*, [online] 25, pp.102109–102109. doi:https://doi.org/10.1016/j.nicl.2019.102109.

Sun, R., Wang, K., Guo, L., Yang, C., Chen, J., Ti, Y. and Sa, Y. (2019). A potential field segmentation based method for tumor segmentation on multi-parametric MRI of glioma cancer patients. *BMC Medical Imaging*, [online] 19(1). doi:https://doi.org/10.1186/s12880-019-0348-y.

Syed Ali Yazdan, Ahmad, R., Iqbal, N., Rizwan, A., Anam Nawaz Khan and Kim, D.-H. (2022). An Efficient Multi-Scale Convolutional Neural Network Based Multi-Class Brain MRI Classification for SaMD. *Tomography*, [online] 8(4), pp.1905–1927. doi:https://doi.org/10.3390/tomography8040161.

Talo, M., Ulas Baran Baloglu, Özal Yıldırım and U Rajendra Acharya (2019). Application of deep transfer learning for automated brain abnormality classification using MR images. *Cognitive Systems Research*, [online] 54, pp.176–188. doi:https://doi.org/10.1016/j.cogsys.2018.12.007.

Tan, M. and Le, Q.V. (2019). *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks*. [online] arXiv.org. Available at: https://arxiv.org/abs/1905.11946 [Accessed 3 Sep. 2024].

Wang, H., Xu, T., Huang, Q., Jin, W. and Chen, J. (2020). Immunotherapy for Malignant Glioma: Current Status and Future Directions. *Trends in Pharmacological Sciences*, [online] 41(2), pp.123–138. doi:https://doi.org/10.1016/j.tips.2019.12.003.

Wu, X., Bi, L., Fulham, M., David Dagan Feng, Zhou, L. and Kim, J. (2021). Unsupervised brain tumor segmentation using a symmetric-driven adversarial network. *Neurocomputing*, [online] 455, pp.242–254. doi:https://doi.org/10.1016/j.neucom.2021.05.073.

Xie, S., Girshick, R., Dollár, P., Tu, Z. and He, K. (2016). *Aggregated Residual Transformations for Deep Neural Networks*. [online] arXiv.org. Available at: https://arxiv.org/abs/1611.05431v2 [Accessed 3 Sep. 2024].

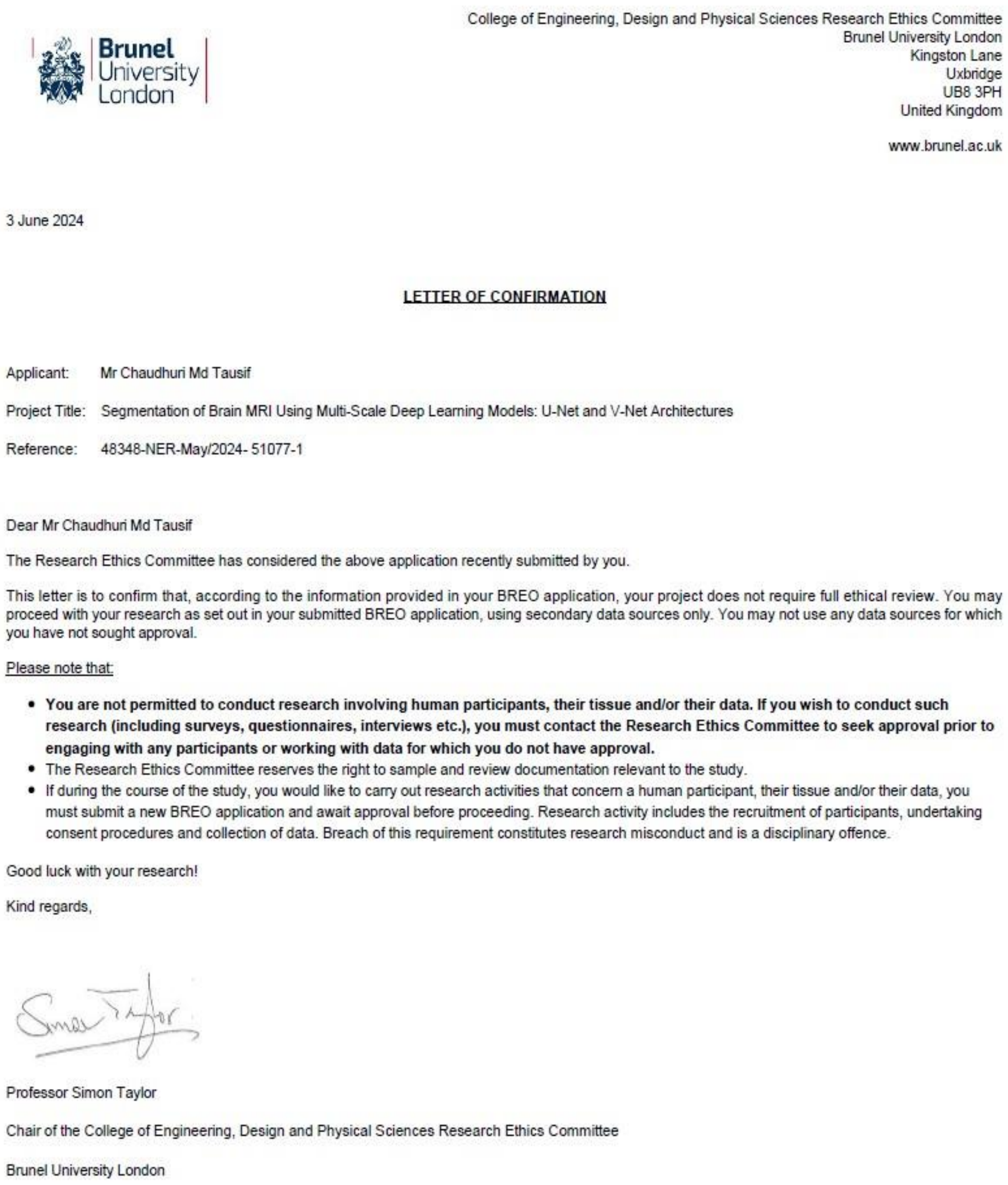
Zhang, J., Zeng, J., Qin, P. and Zhao, L. (2021a). Brain tumor segmentation of multi-modality MR images via triple intersecting U-Nets. *Neurocomputing*, [online] 421, pp.195–209. doi:https://doi.org/10.1016/j.neucom.2020.09.016.

Zhang, Y., Lu, Y., Chen, W., Chang, Y., Gu, H. and Yu, B. (2021b). MSMANet: A multi-scale mesh aggregation network for brain tumor segmentation. *Applied Soft Computing*, [online] 110, pp.107733–107733. doi:https://doi.org/10.1016/j.asoc.2021.107733.

Zhou, X., Li, X., Hu, K., Zhang, Y., Chen, Z. and Gao, X. (2021). ERV-Net: An efficient 3D residual neural network for brain tumor segmentation. *Expert Systems with Applications*, [online] 170, pp.114566–114566. doi:https://doi.org/10.1016/j.eswa.2021.114566.

Zhou, Z., He, Z. and Jia, Y. (2020). *AFPNet: A 3D Fully Convolutional Neural Network with Atrous-convolution Feature Pyramid for Brain Tumor...* [online] ResearchGate. Available at: https://www.researchgate.net/publication/340552411_AFPNet_A_3D_Fully_Convolutional_Neural_Network_with_Atrous-convolution_Feature_Pyramid_for_Brain_Tumor_Segmentation_via_MRI_Images [Accessed 1 Sep. 2024].

Appendix A: ETHICAL APPROVAL



Brunel University London

College of Engineering, Design and Physical Sciences Research Ethics Committee
Brunel University London
Kingston Lane
Uxbridge
UB8 3PH
United Kingdom

www.brunel.ac.uk

3 June 2024

LETTER OF CONFIRMATION

Applicant: Mr Chaudhuri Md Tausif

Project Title: Segmentation of Brain MRI Using Multi-Scale Deep Learning Models: U-Net and V-Net Architectures

Reference: 48348-NER-May/2024- 51077-1

Dear Mr Chaudhuri Md Tausif

The Research Ethics Committee has considered the above application recently submitted by you.

This letter is to confirm that, according to the information provided in your BREO application, your project does not require full ethical review. You may proceed with your research as set out in your submitted BREO application, using secondary data sources only. You may not use any data sources for which you have not sought approval.

Please note that:

- **You are not permitted to conduct research involving human participants, their tissue and/or their data. If you wish to conduct such research (including surveys, questionnaires, interviews etc.), you must contact the Research Ethics Committee to seek approval prior to engaging with any participants or working with data for which you do not have approval.**
- The Research Ethics Committee reserves the right to sample and review documentation relevant to the study.
- If during the course of the study, you would like to carry out research activities that concern a human participant, their tissue and/or their data, you must submit a new BREO application and await approval before proceeding. Research activity includes the recruitment of participants, undertaking consent procedures and collection of data. Breach of this requirement constitutes research misconduct and is a disciplinary offence.

Good luck with your research!

Kind regards,

Simon Taylor

Professor Simon Taylor

Chair of the College of Engineering, Design and Physical Sciences Research Ethics Committee

Brunel University London

The PDF file of the Ethics Letter will be uploaded as a zip file named CS5500_2355912 in appendix folder of Wiseflow.

Appendix B: Code

Here I have given the Kaggle code and dataset links and also my GitHub links.

GitHub Links –

https://github.com/CMDTausif/Brain-MRI-Segmentation-using-PSPNET-and-UNET-EfficientNetB7

Kaggle Dataset Link - https://www.kaggle.com/datasets/mateuszbuda/lgg-mri-segmentation

Kaggle Codes Link -

1. PSPNet From Scratch V1 –

https://www.kaggle.com/code/chaudhurimdtausif/implementation3-v4

2. PSPNet From Scratch V2 –

https://www.kaggle.com/code/chaudhurimdtausif/pspnet-from-scratch

3. PSPNet with ResNeXt50_32x4d-

https://www.kaggle.com/code/chaudhurimdtausif/psp-net-pytorch

4. UNet with EfficientNetB7 –

https://www.kaggle.com/code/chaudhurimdtausif/efiicientnetb7-v2

The codes will be uploaded as a zip file named CS5500_2355912 in the appendix folder of Wiseflow.