

UNIVERSITY OF BAHRAIN

College of Information Technology



Detection and Classification of Brain Tumor using Deep Transfer Learning Approach

A Thesis Submitted in Partial Fulfillment of the Requirements for the
**MSc. Degree in Machine Learning and Computational
Intelligence**

Submitted by:

Maleeha Muzafar Ali Zulfiqar Mohammad Ismail
202100104

Supervised by:

Prof. Alauddin Al-Omary
(Professor)
University of Bahrain

Kingdom of Bahrain

December 2024



Dissertation Deposit Form

Author Name:	Maleeha Muzafar		
Email:	202100104@stu.uob.edu.bh	Contact No:	33276672
College:	Information Technology		
Department:	Computer Science		
Thesis Title:	Detection and Classification of Brain Tumor using Deep Transfer Learning Approach		
Year:	2024		

Author's Declaration: I agree to the following conditions:

The above thesis will be available (in the University of Bahrain library and externally) and reproduced as necessary at the discretion of the University of Bahrain. It may also be digitized by the University of Bahrain and made available via the university's repository or via other parties on the Internet.

This condition shall apply to ALL copies including electronic copies.

The above thesis has been provided on the understanding that it is copyright protected material and that no quotation from this thesis may be published without proper acknowledgement.

Signature of the Author:		Date:	09-Dec-24
--------------------------	--	-------	-----------

Defense committee's approval

DECLARATION

I declare that this work is the result of my own investigation and that it has not already been accepted in substance for any degree, nor is it currently submitted for any degree.

Signed:  Date: 9/12/2024

MALEEHA MUZAFAR ALI ZULFIQAR

The Thesis defense committee considers the thesis titled:

Detection and Classification of Brain Tumor using Deep Transfer Learning Approach

Submitted by MALEEHA MUZAFAR ALI ZULFIQAR to the College of Information Technology is satisfactory and acceptable for the Master of Science in Machine Learning and Computational Intelligence.

The Defense Committee:



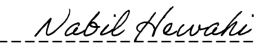
Prof. Alauddin Al-Omary
(Supervisor)

Department of Computer Engineering
University of Bahrain



Prof. Abdulnasir Hossen
(External Examiner)

Sultan Qaboos University



Prof. NABIL HEWAHI
(Internal Examiner)

Department of Computer Science
University of Bahrain

Abstract

Precise and timely recognition of brain tumor is crucial for treatment planning to prolong the life span of the patient. Over the past decades many research efforts have been focused on classification and detection of brain tumors. However, it is a daunting research problem due to the tumor being different in shape, size, and location. Magnetic Resonance Imaging (MRI) is a commonly prescribed diagnostic test for diagnosing brain tumors. This study investigates the use of EfficientNetV2S, a pre-trained model, to diagnose brain tumors. Furthermore, the placement and order of adding layers in the top final classifier was also studied. Data augmentation was incorporated to expand the dataset size. Finally fine-tuning was applied on the best performing model in the previous experiment to improve the performance and accuracy of the model. The tumors were categorized into four classes: glioma, meningioma, pituitary, and no-tumor. The final classification layer included a dense layer along with the SoftMax activation function. It was concluded that the highest performing model was the fine-tuned EfficientNetV2S model with no pooling layer and batch normalization layer added after the dense layer. The proposed model was able to score 99.98% in training f1-score, while scoring 99.47% in testing f1-score. It was also concluded that the placement and order in which layers are in the final classifier has an impact on the model performance.

Keywords: *brain tumor classification, EfficientNetV2S, MRI, BraTS, transfer learning, fine-tuning*

Acknowledgment

I am grateful to Almighty God for all the strength and knowledge that He blessed me with to complete this research. Without His help and the constant prayers of my parents, I doubt I would be able to finish this study.

I want to sincerely thank my supervisor, Dr. Alauddin Al-Omary, for his constant support, help, and guidance. His encouragement and dedication have been influential in completing this thesis. He has been the reason I developed an interest in computer vision. His course was extremely beneficial in giving me a direction, focus, and passion for pursuing this thesis topic.

Finally, I am always indebted to my loving family for always being there through thick and thin and standing by my side with their unwavering support while I complete this journey in increasing my knowledge.

Table of Contents

Defense committee’s approval.....	C
Abstract.....	D
Acknowledgment.....	E
Table of Contents.....	F
List of Tables.....	H
List of Figures.....	I
List of Abbreviations.....	K
Chapter 1: Introduction.....	1
1.1 Overview.....	1
1.2 Research Problem	4
1.3 Research Questions.....	4
1.4 Research Objectives	5
1.5 Research Significance.....	5
1.6 Research Contribution	6
1.7 Thesis Outline.....	6
Chapter 2: Literature Review	8
2.1 Introduction.....	8
2.2 Related Work	8
2.3 Gaps in Previous Research	12
Chapter 3: Methodology.....	15
3.1 Introduction.....	15
3.2 Dataset	15
3.3 Data Preprocessing.....	16
3.4 Data Augmentation.....	17
3.5 Model with Transfer Learning	18
3.5.1 MBConv	20
3.5.2 SE Operation.....	20
3.5.3 Fused MBConv	22
3.6 Hyperparameter Selection	23

3.7 Pooling Layers	23
3.7.1 Max Pooling Layer.....	23
3.7.2 Global Average Pooling Layer.....	24
3.8 Batch Normalization Layer	24
3.9 Dense Layer	25
3.10 Dense Layer with Softmax Activation – Output Layer	25
3.11 Activation Function	25
3.12 Convolution.....	26
3.13 Model Training.....	26
3.14 Model Optimization with Finetuning	27
3.15 Proposed Architecture and Experiments.....	27
Chapter 4: Result.....	28
4.1 Introduction.....	28
4.2 Metrics Used	28
4.3 Model Evaluation.....	30
4.3.1 Experiment 1: No Pooling Layer	30
4.3.2 Experiment 2: Global Average Pooling Layer	33
4.3.3 Experiment 3: Max Pooling Layer	36
4.3.4 Experiment 4: Batch Normalization before Dense Layer	39
4.3.5 Experiment 5: Batch Normalization after Dense Layer	42
4.3.6 Experiment 6: Finetuning the best classifier.....	45
4.4 Discussion	48
Chapter 5: Conclusion, Limitations, and Future Work.....	49
5.1 Introduction.....	49
5.2 Study Limitations	49
5.3 Conclusion	50
5.4 Future Work	51
References	52
ملخص البحث	56

List of Tables

Table 2.3.1: Literature review summary	13
Table 3.2.1: Dataset organization.....	16
Table 3.6.1: Selected hyperparameters	23
Table 4.3.1: Experiment 1 metrics - no pooling layer experiment.....	30
Table 4.3.2: Experiment 2 metrics - global average pooling layer experiment.....	33
Table 4.3.3: Experiment 3 metrics - max pooling layer experiment	36
Table 4.3.4: Experiment 4 metrics - batch normalization before dense layer	39
Table 4.3.5: Experiment 5 metrics - batch normalization after dense layer.....	42
Table 4.3.6: Experiment 6 metrics - finetuning.....	45

List of Figures

Figure 1.1.1: Benign and malignant brain tumor as seen in MRI.....	1
Figure 1.1.2: Meningioma, glioma, and pituitary tumors in different visual rotations of MRI.....	3
Figure 3.3.1: MRI scans from the two datasets.....	17
Figure 3.4.1: Original vs. augmented MRI generated by ImageDataGenerator.....	18
Figure 3.4.2: ImageDataGenerator workflow	18
Figure 3.5.1: EfficientNetV2-S Architecture	19
Figure 3.5.2: SE operation (Tan & Le, EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks, 2020).....	21
Figure 3.5.3: MBConv vs. FusedMBConv (Tan & Le, EfficientNetV2: Smaller Models and Faster Training, 2021).....	22
Figure 3.7.1: Max pooling layer representation	24
Figure 3.13.1: Visual diagram for the proposed model workflow (Incir & Bozkurt, 2024)	26
Figure 4.3.1: Experiment 1 architecture - no pooling layer	30
Figure 4.3.2: Experiment 1 - Loss Accuracy Curve - No Pooling.....	31
Figure 4.3.3: Experiment 1 - Confusion Matrix - No Pooling (Test DS1).....	32
Figure 4.3.4: Experiment 1 - Confusion Matrix - No Pooling (Test DS2).....	32
Figure 4.3.5: Experiment 2 architecture - global average pooling layer	33
Figure 4.3.6: Experiment 2 - Loss accuracy curve - Global Average Pooling.....	34
Figure 4.3.7: Experiment 2 - Confusion Matrix - Global Average Pooling (Test DS1).....	35
Figure 4.3.8: Experiment 2 - Confusion Matrix - Global Average Pooling (Test DS2).....	35
Figure 4.3.9: Experiment 3 architecture - max pooling layer.....	36
Figure 4.3.10: Experiment 3 - Loss accuracy curve - Max Pooling	37
Figure 4.3.11: Experiment 3 - Confusion Matrix - Max Pooling (Test DS1)	38
Figure 4.3.12: Experiment 3 - Confusion Matrix - Max Pooling (Test DS2)	38
Figure 4.3.13: Experiment 4 architecture - batch normalization before dense layer	39
Figure 4.3.14: Experiment 4 - Loss accuracy curve - batch normalization before dense layer...40	
Figure 4.3.15: Experiment 4 - Confusion Matrix - batch normalization before dense layer (Test DS1)	41
Figure 4.3.16: Experiment 4 - Confusion Matrix - batch normalization before dense layer (Test DS2)	41
Figure 4.3.17: Experiment 5 architecture - batch normalization after dense layer.....	42

Figure 4.3.18: Experiment 5 - Loss accuracy curve - batch normalization after dense layer	43
Figure 4.3.19: Experiment 5 - Confusion Matrix - batch normalization after dense layer (Test DS1)	44
Figure 4.3.20: Experiment 5 - Confusion Matrix - batch normalization after dense layer (Test DS2)	44
Figure 4.3.21: Experiment 6 architecture - finetuning.....	45
Figure 4.3.22: Experiment 6 – Loss accuracy curve - finetuning	46
Figure 4.3.23: Experiment 6 - Confusion Matrix - finetuning (Test DS1)	47
Figure 4.3.24: Experiment 6 - Confusion Matrix - finetuning (Test DS2).....	47

List of Abbreviations

ADAM	Adaptive Moment Estimation
AUC	Area under the Receiver Operating Characteristics Curve
AWS	Amazon Web Services
CLAHE	Contrast Limited Adaptive Histogram Equalization
CNN	Convolutional Neural Network
CT	Computed Tomography
DCNN	Deep Convolutional Neural Network
DS1	Dataset 1
DS2	Dataset 2
FCL	Fully Connected Layer
GLCM	Gray-Level Co-Occurrence Matrix
GAN	Generative Adversarial Network
MRI	Magnetic Resonance Image
PET	Positron Emission Tomography
PCA	Principal Component Analysis
ReLU	Rectified Linear Unit
ROC	Receiver Operating Characteristics
WHO	World Health Organization
YOLO	You Only Look Once

Chapter 1: Introduction

1.1 Overview

Brain Tumor is amongst the deadliest diseases known to mankind. Brain tumors are caused by the intense proliferation and multiplication of abnormal brain cells. As the number of abnormal cells increase, the pressure within the patient's brain also increases. This increase in pressure can be life-threatening (Brain Tumors, 2024).

There are two primary classifications of brain tumors, viz, benign (non-cancerous), and malignant (cancerous). Benign brain tumors grow gradually without invading other parts of the body. However, malignant brain tumors are generally invasive as they proliferate rapidly and destroy the surrounding tissues in the patient's brain (Brain Tumors, 2024). In Figure 1.1.1, the scan on the left depicts a benign tumor, while the scan on the right depicts a malignant tumor. Because of the aggressive nature of the disease, early recognition of brain tumor is a critical problem in the medical field.

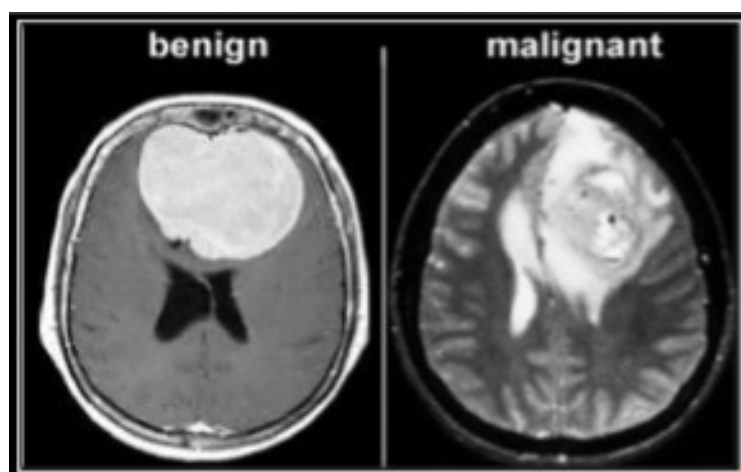


Figure 1.1.1: Benign and malignant brain tumor as seen in MRI

Brain tumors can further be categorized into primary and secondary tumors. Primary tumors can either develop within the patient's spinal cord or the brain

(Drevelegas, 2010). Based on the tumor's potential of multiplying and its aggressiveness, primary tumors can be classified into high-grade (malignant) and low-grade tumors (benign). However, secondary tumors, also referred to as metastatic brain tumors, develop in various other body parts but are eventually spread in the patient's brain. Secondary tumors are generally more common than primary tumors.

Glioma tumor is a primary tumor that develops in the brain's glial cells or the spinal cord. Glial cells are responsible for the regular working of the nervous system as they nourish the neurons in the brain. Glioma tumor can be further classified into different types depending on the origin of the tumor cell. This is the most aggressive kind of brain tumor because the disease has a life expectancy of 12 – 15 months for the most malignant form of glioma (Goodenberger & Jenkins, 2012).

Meningioma is a slow-growing tumor that develops in brain's protective membranes called meninges. Meninges is responsible for covering and protecting the brain and the spinal cord. Meningioma is generally considered a benign tumor, but it can take the form of a malignant tumor in rare cases (Osborn, Louis, Poussaint, Linscott, & Salzman, 2022). Meningioma is frequently diagnosed in patients over the age of 60 and the cause of this kind of tumor is currently unknown.

Pituitary tumor originates within the pituitary gland. The tumors found in the pituitary gland are normally benign tumors and can be further categorized based their production of hormones and their size. Smaller pituitary tumors generally do not trigger any symptoms; however, larger pituitary tumors may exert pressure on the brain's adjacent structure (Osborn, Louis, Poussaint, Linscott, & Salzman, 2022).

A number of biomedical imaging techniques are utilized in diagnosing brain tumors. Some of these techniques include Computed Tomography (CT), Magnetic Resonance Imaging (MRI), Positron Emission Tomography (PET), Single Photon Emission Computed Tomography (SPECT), etc. (Drevelegas, 2010). With the advancement in technology, these imaging techniques can provide precise information about the size, along with the location, and grade of the tumor, if present. MRI is the most commonly prescribed diagnostic test by medical professionals due to its non-invasive nature. It makes use of radio waves combined with powerful magnetic fields to generate precise images of the brain core (Litjens, 2017).

This study focuses on utilizing MRI scans for the detection and classification of three types of brain tumors: *glioma* (malignant), *meningioma* (benign), *pituitary* (benign), along with a *no-tumor* class to depict a healthy MRI scan. Figure 1.1.2 shows MRI scans for meningioma, glioma, and pituitary tumors on different planes as are used by radiologists.

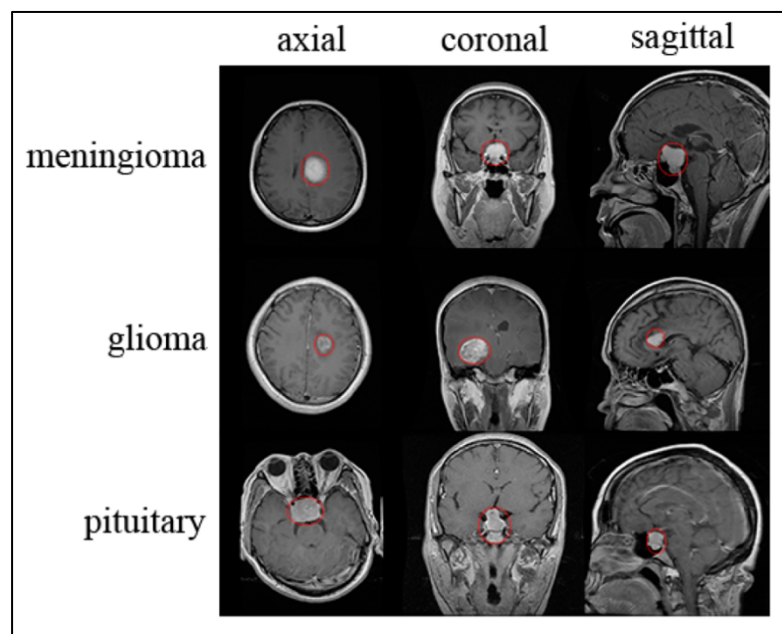


Figure 1.1.2: Meningioma, glioma, and pituitary tumors in different visual rotations of MRI

1.2 Research Problem

Early and precise prognosis of brain tumor is vital for medical experts to create treatment and recovery plans for their patients. Early diagnosis is key in preventing the prognosis of brain tumors. However, currently, manual radiological analysis and assessment are used for diagnosing patients which introduces significant delay in the commencement of treatment (Anaya Isaza, Mera Jiménez, Verdugo Alejo, & Sarasti, 2023). While there exist many methods for brain tumor diagnosis via assistance of machine learning along with deep learning, there is always the need to enhance the diagnostic accuracy and develop more precise, efficient, and trustworthy solutions for brain tumor classification (Turner & Nancy, 2022). Various pre-trained deep learning models have been studied in this problem domain. This research revolves around studying the use of deep transfer learning in the domain of brain tumor recognition with an aim to further enhance its diagnostic accuracy. A pre-trained model is used to detect and classify brain tumor in MRI scans. Moreover, data augmentation is applied to increase the size of the datasets used in the study. Finally, the effect of pooling layers, batch normalization layers and finetuning is studied too.

1.3 Research Questions

This study targets to answer and study the undermentioned questions:

1. **What is the optimal architecture for utilizing pre-trained deep learning models for brain tumor classification?** This question addresses the architecture of the model's final classifier. It aims to study the effect, placement, and absence of different layers in the final classifier and its impact on the accuracy of the model. In other words, the performance of the model with and without pooling layers is evaluated. Also, the performance of the model with batch normalization before and after the dense layer is evaluated.

2. **How does combining transfer learning along with data augmentation and fine-tuning impact the efficiency and accuracy of the pre-trained model?** Data augmentation helps in expanding the size of the dataset. This question aims at studying the effect data augmentation and fine-tuning can have on the transfer learning model.
3. **Can fine-tuned pre-trained model that is trained on one dataset be applied on different datasets?** The generalization capability of the fine-tuned model is explored in this question. This includes testing the final model on one dataset, while it was trained on a different dataset.

1.4 Research Objectives

The primary focus of this thesis is to study existing work done in the field of brain tumor detection and strive to refine the performance and results of the existing work done. All the key objectives are as follows:

1. Investigate the work of other researchers in the domain.
2. Implement data augmentation to increase the size of the MRI dataset.
3. Use pre-trained model for classifying brain tumor in biomedical images.
4. Investigate performance of pooling and batch normalization layers.
5. Investigate the use of transfer learning and fine-tuning to improve the diagnostic accuracy.

1.5 Research Significance

The World Health Organization (WHO) published in their world cancer report that brain cancer usually results in critical complications for patients and causes increased morbidity (Sultan, Salem, & Al-Atabany, 2019). In the United Kingdom, the cancer research corporation published that 5250 deaths occur annually due to brain and other intracranial tumors (Bakas, et al., 2021). Significant advancements have been made in deep learning and biomedical

imaging techniques. An automated solution for classifying brain tumor using computer-aided technology can significantly reduce the time it takes for medical experts to diagnose deadly diseases. In this research, the potential of deep transfer learning to classify MRI scans with brain tumors is studied.

1.6 Research Contribution

This study focuses on utilizing MRI scans for detecting and classifying three kinds of tumors of the brain: glioma (malignant), meningioma (benign), pituitary (benign), and no-tumor to help medical professionals enhance the speed and diagnostic accuracy in brain tumor detection. The study also focuses on investigating the performance of pre-trained EfficientNetV2S model to diagnose brain tumors with different pooling layers in the top final classifier. The objective is to explore how the performance of the pre-trained EfficientNetV2S model can be enhanced by using pooling layers, batch normalization layers, and fine-tuning.

1.7 Thesis Outline

This thesis is organized into multiple chapters as follows:

- **Chapter 1** comprises of an introduction to the study. It also includes details on the research significance along with the research problem, questions, and objectives tackled in this study.
- **Chapter 2** comprises of the literature review that examines and reviews the recent work done in the domain of brain tumor classification using deep transfer learning and highlights the research gaps in the previous studies.
- **Chapter 3** sheds light on the implementation carried out along with the methodology applied to solve the research problem. Experimental setup is discussed along with the data collection, pre-processing, training, and validation.

- **Chapter 4** presents the results and findings along with the evaluation of the proposed model's performance and workflows. Furthermore, analysis is done on how different factors affect the model's performance.
- **Chapter 5** concludes the study, summarizes the findings, highlights the contributions, stipulates the limitations, and recommends further research advancements in the domain.

Chapter 2: Literature Review

2.1 Introduction

Firstly, in recent years, deep transfer learning techniques has garnered attention in classifying biomedical images. The most studied biomedical imaging modality is MRI. MRI which is commonly used in diagnosing various diseases. Numerous approaches and methods have been proposed for brain tumor classification using MRI scans. With advancements in computer vision, researchers have been experimenting with the use of pre-trained models in classifying brain tumors. This literature review explores the comprehensive research on classification of brain tumor using deep transfer learning approaches. By integrating and investigating a diverse list of studies, the aim of this literature review is to map the progression of methodologies, the efficiency of different deep learning models, and the fundamental aspects aiding to accurate and successful brain tumor classification.

2.2 Related Work

(Alla & Athota, 2022) investigated the use of transfer learning and fine-tuning in detecting tumor in MRI images. One public dataset was acquired from Kaggle with 3000 images. All images were resized and converted to vectors and grayscale. Data augmentation techniques were incorporated to expand the experimental dataset. Every image was augmented 16 times by flipping, rotating, and zooming the original image. Three CNN models, viz, VGG-16, VGG-19, and Inception-V3 were experimented. VGG-19 achieved the highest prediction accuracy (97%) followed by VGG-16 (96%) and Inception-V3 (89%). This study revealed that experiments performed with fine-tuning achieve higher accuracy when compared to experiments conducted with transfer learning alone.

(Tenghongsakul, Kanjanasurat, Archevapanich, Purahong, & Lasakul, 2023) presented the comparison of four pre-trained models, viz, InceptionResNet-V2,

MobileNet-V2, ResNet50, and VGG-16. The study aimed at identifying the most suitable CNN model for detecting brain tumor in MRI images. A public dataset was acquired from Kaggle with 253 images. All images were resized, augmented, and normalized. Contrast Limited Adaptive Histogram Equalization (CLAHE) was experimented for improving the image quality and contrast. Researchers compared three optimizers in the study, namely, Adam, SDG, and RMSProp. The experiment was run for 30 epochs and VGG-16 using CLAHE and RMSprop obtained 100% in all metrics, viz, F1-score, recall, precision, and accuracy.

(Shelatkar, Urvashi, Shorfuzzaman, Alsufyani, & Lakshman, 2022) used Brats 2021 dataset to identify and classify brain tumors using different variants of YOLOv5 (You Only Look Once) model. Transfer learning and finetuning were investigated using the pre-trained weights via the cocoa model. The researchers were able to achieve a precision of 88%. However, the research did not implement any form of data augmentation or cross validation to address the issues of model overfitting due to the smaller size of the dataset used.

(Modiya & Vahora, 2022) proposed the use of the classifiers EfficientnetB7 and VGG-16 to investigate the usage of transfer learning in detection of brain tumors along with the significance of PCA (Principal Component Analysis). A public dataset was acquired from Kaggle with 3000 images. All images were resized and augmented to increase the dataset size. Furthermore, application of PCA for dimensionality reduction was investigated. For the results, EfficientnetB7 coupled with PCA performed better with a prediction accuracy of 80%. Without PCA, the classifier resulted in an accuracy of 63%.

(Talukder, et al., 2022) aimed to explore the usage of transfer learning along with finetuning to detect and classify different types of brain tumors. The study experimented on 3,064 brain MRI images available on Figshare MRI. For data preprocessing, all images were resized, sharpened, scaled, and labeled. All images

were augmented to expand the dataset and to avoid overfitting. The approach employed various pre-trained models, including InceptionResNetV2, Xception, DenseNet201, and ResNet50V2. For fine-tuning, Global Average Pooling2D, Batch Normalization, and Dense layer were added to the suggested model. It was discovered that ResNet50V2 outperformed the other classifiers by achieving an accuracy of 99.68%.

(Kumar, Prasad, & Metan, 2022) presented ResNet 152 Transfer Learning model for detection and classification brain tumors. The study used 280 MRI images from the Brats dataset. Otsu Binarization method was used for preprocessing the input images. Feature extraction was applied using Gray-Level Co-Occurrence Matrix methods. All of the batch normalization layer present in the model was discarded. Input images were classified into two classes: benign and malignant. Furthermore, the parameters of Hyb-DCNN-ResNet 152 TL were fine-tuned with the use of Covid-19 optimization. Their proposed architecture was able to achieve an accuracy of 97.28%.

(Srinivas, et al., 2021) presented a comparative study of three pre-trained CNN model, viz, Inception-V3, VGG-16, and ResNet-50 to predict tumor in 256 brain MRI images. Image preprocessing and augmentation were applied to avoid overfitting problems. All the convolution layers were frozen for the experiments. The results demonstrated that VGG-16 performed the best with an accuracy of 88.26%.

In another study, researchers utilized the EfficientNetV2 model for the brain tumor classification problem (Wastika, Amda, Sihono, & Prasetyo, 2023). The dataset utilized consisted of brain tumor MRI scans grouped into no-tumor, glioma, pituitary, and meningioma. The architecture's final layer was updated to adapt to the new classes. Adam optimizer along with categorical cross entropy loss function were incorporated in the setup. The experiments were run on the B0, B1,

B2, and B3 variants of EfficientNetV2 model. Researchers used the Crystal Clean dataset and were able to record accuracies greater than 99% for all the investigated models.

Pre-trained ResNet50 network was studied by (Anaya Isaza & Jiménez, 2022) for brain tumor detection problem. In their research, the implemented data augmentation to expand the dataset used in the experiment since it originally contained 110 images only. Principal component analysis (PCA) was also implemented for dimensionality reduction. K-fold validation was implemented using a total of 10 folds. For the hyperparameters, they opted for binary cross-entropy loss function, Adadelta optimizer along with a batch size of 10 and training epochs of 50. The experiment resulted in an f1-score of 92.4%.

(Chattopadhyay & Maitra, 2022) agreed that manually assessing a huge number of MRI scans by medical experts can be prone to human error, and therefore, proposed a 9-layer CNN architecture. The 2020 BraTS dataset was used to run the experiments. The research included comparing the performance of the final layer activation functions and optimizers. The comparison was made between SVM, sigmoid, and softmax activation functions and between Adamax and RMSProp optimizers. With just 11 epochs, researchers were able to produce an output accuracy of 97.94% using softmax activation function and RMSProp optimizer. However, the research did not implement any form of data augmentation or cross validation to address the issues of model overfitting due to the smaller size of the dataset used. The researchers did mention that some of the images were deleted from the dataset to curb the model from overfitting.

(Islam, et al., 2023) studied the performance of four deep transfer learning models in brain tumor diagnosis. The architectures included: InceptionV3, DenseNet121, VGG19, and MobileNet. To validate the models, researchers used the SARTAJ dataset along with Figshare and Br35H datasets. Image augmentation was applied

to balance out all the classes. According to their results, MobileNet outperformed the other architectures with an accuracy of 99.39%.

(Dhakshnamurthy, Govindan, Sreerangan, Nagarajan, & Thomas, 2024) experimented with AlexNet, VGG16, and ResNet-50 in brain tumor diagnosis. The dataset used in this study was sourced from Kaggle. The name of the dataset was not specifically mentioned; however, it contained a total of 3264 MRI scans which was further sliced up into training and testing sets. VGG-16 was the highest performing classifier with an accuracy of 97.66% and an f1 score of 97.62%. The researchers further studied the performance of a hybrid model amalgamating VGG-16 and ResNet50 models which eventually outperformed all the other classifiers used in the study with an accuracy and f1 score of 99.98%.

(Incir & Bozkurt, 2024) applied six different deep learning algorithms to experiment with the brain tumor classification problem. These included EfficientNetV2-M, ResNet50, MobileNet, VGG16, DenseNet121, and InceptionV3. A publicly available dataset on Kaggle was utilized in this study. They applied random rotation technique to augment the dataset for the study. Their result showed that EfficientNetV2-M outperformed all other classifiers with an accuracy of 98.01%.

2.3 Gaps in Previous Research

This literature review was targeted at addressing the underrepresented areas and gaps in the brain tumor classification domain. The review consisted of an extensive search of studies done within the domain in the recent past.

The review highlighted the importance of having additional advancements within the domain to aid medical professionals in diagnosing brain tumor. More and more studies were targeted at improving the performance and increasing the efficiency of pre-trained models in brain tumor classification.

However, various gaps and limitations were identified while exploring the existing literature. One such predominant gap was the comparative study of the layers in the final classifier of all the pre-trained models. These could include pooling layers and batch normalization layer. There were some studies that discussed and compared the effects and performance of the different hyperparameters of the models. However, almost none of the literature investigated the effects of pooling layers and batch normalization layer in the final classifier of the pre-trained models.

The results of all the studied literature are summarized in Table 2.3.1.

Table 2.3.1: Literature review summary

Literature	Dataset	Model	Accuracy	Gaps
(Alla & Athota, 2022)	Br35H	InceptionResNet-V2	96%	
		MobileNet-V2	97%	
		ResNet50	89%	
		VGG-16	89%	
(Tenghongsakul, Kanjanasurat, Archevapanich, Purahong, & Lasakul, 2023)	Br35H	CLAHE + VGG-16	100%	No mention of the final layers used / their organization
		ResNet50	97%	
		Inception-V3	98.08%	
		MobileNet-V2	99.90%	
(Shelatkar, Urvashi, Shorfuzzaman, Alsufyani, & Lakshmana, 2022)	Brain MRI Images for Brain Tumor Detection	YOLOv5	88%	Small dataset, no augmentation
(Modiya & Vahora, 2022)	Figshare brain tumor dataset	EfficientNetB7 + PCA	80%	Learning curves not included, result analysis is difficult
		EfficientNetB7	63%	
		VGG-16 + PCA	69%	
		VGG-16	56%	
(Talukder, et al., 2022)	Brain magnetic resonance imaging tumor dataset	InceptionResNetV2	99.36%	
		Xception	98.40%	
		DenseNet201	98.72%	
		ResNet50V2	99.68%	

Literature	Dataset	Model	Accuracy	Gaps
(Kumar, Prasad, & Metan, 2022)	BraTS 2018	ResNet152	97.28%	small dataset, no augmentation, may suffer from overfitting
(Srinivas, et al., 2021)	Kaggle	Inception-V3	78%	
		VGG-16	88.26%	
		ResNet-50	78%	
(Wastika, Amda, Sihono, & Prasetyo, 2023)	Figshare brain tumor dataset	EfficientNetV2B0	99.43%	
		EfficientNetV2B1	99.37%	
		EfficientNetV2B2	99.17%	
		EfficientNetV2B3	99.22%	
(Anaya Isaza & Jiménez, 2022)	TCGA-LGG	ResNet50	92.34%	
(Chattopadhyay & Maitra, 2022)	BraTS 2018	CNN	99.74%	small dataset, no augmentation, may suffer from overfitting
(Islam, et al., 2023)	Brain Tumor MRI Dataset	VGG19	96.72%	
		InceptionV3	97.10%	
		DenseNet121	98.32%	
		MobileNet	99.39%	
(Dhakshnamurthy, Govindan, Sreerangan, Nagarajan, & Thomas, 2024)	Kaggle	AlexNet	95.60%	Missing dataset name and reference, final classifier layers not mentioned
		VGG16	97.66%	
		ResNet50	96.90%	
		VGG16 + ResNet50	99.98%	
(Incir & Bozkurt, 2024)	Brain Tumor Classification (MRI)	EfficientNetV2-M	98.01%	Final classifier layers used not mentioned, only one data augmentation technique was applied
		ResNet50	96.82%	
		MobileNet	97.22%	
		VGG16	97.35%	
		DenseNet121	97.88%	
		InceptionV3	97.75%	

Chapter 3: Methodology

3.1 Introduction

This chapter provides details on the proposed methodologies implemented in this study. The key objective of this study is to enhance the diagnostic accuracy of detecting and classifying brain tumor in MRI using deep transfer learning. The chapter is broken into different sections that detail every phase of the experiment in chronological order.

Firstly, to conduct the experiment in this research, an experimental setup was created. Amazon Web Services (AWS) SageMaker Studio account was setup to run the experiments. SageMaker Studio is a fully integrated development environment for machine learning. Using the Sagemaker Studio, a JupyterLab instance of ml.g4dn.xlarge was run to accommodate the usage of GPU for model training. The ml.g4dn.xlarge instance provides 16 GiB instance memory, 1 NVIDIA T4 GPU, and 16GB GPU memory. To run experiments in computer vision, it is crucial to use GPU for lightning performance.

3.2 Dataset

In this study, two datasets were used. To train the model, Brain Tumor Segmentation (BraTS) dataset (Bakas, et al., 2021). BraTS dataset is a publicly available dataset comprising of 2D MRI scans of three types of brain tumors: glioma, meningioma, pituitary, along with a class for the no tumor (healthy scan) category. However, this dataset originally consists of images in the format of dicom/nii files. This format is used to store medical images. Due to the large size of the files in .dicom/.nii format, a JPG version of the dataset was sourced from Kaggle instead (Rao, 2021). These images in the dataset are available in three different plane views: sagittal, axial, and coronal. All the MRI scans are partitioned into their respective class folders which are then further fragmented into training

and testing folders. The training set consists of 5,712 MRI scans whereas the testing set consists of 1,311 organized as depicted in Table 3.2.1. This dataset was used as the finetuned dataset.

The second dataset used in this study for testing purposes was also sourced from Kaggle (Bhuvaji, Kadam, Bhumkar, Dedge, & Kanchan, 2020). It is called Brain Tumor Classification (MRI) and is publicly available for research purposes. All images are categorized in their respective class folders. The dataset is also organized into two distinct collections, viz, testing and training set. About 25% of samples from the training set were used as validation set. Overall, in total, the training set consists of 2,870 scans, whereas the testing set consists of 394 MRI scans. The organization of this dataset is depicted in Table 3.2.1.

Table 3.2.1: Dataset organization

Class	BraTS		Brain Tumor Classification (MRI)	
	Train	Test	Train	Test
Glioma	1321	300	826	100
Meningioma	1339	306	822	115
Pituitary	1595	405	395	74
No tumor	1457	300	827	105

3.3 Data Preprocessing

The initial step in this experiment involves preprocessing of the data. Preprocessing data helps enhance the quality of the dataset used in the investigation. The first preprocessing technique used in this study is image resizing. All images in the dataset were resized to 224 x 224 size as this was the most used size in all the other studies. Moreover, this high resolution could help capture more features when compared to an input with smaller resolution. This was done to firstly optimize the memory while running the experiment and secondly to match the input requirements of the pre-trained model intended to be

investigated in this research. Some samples for every class in the dataset are shown in Figure 3.3.1.

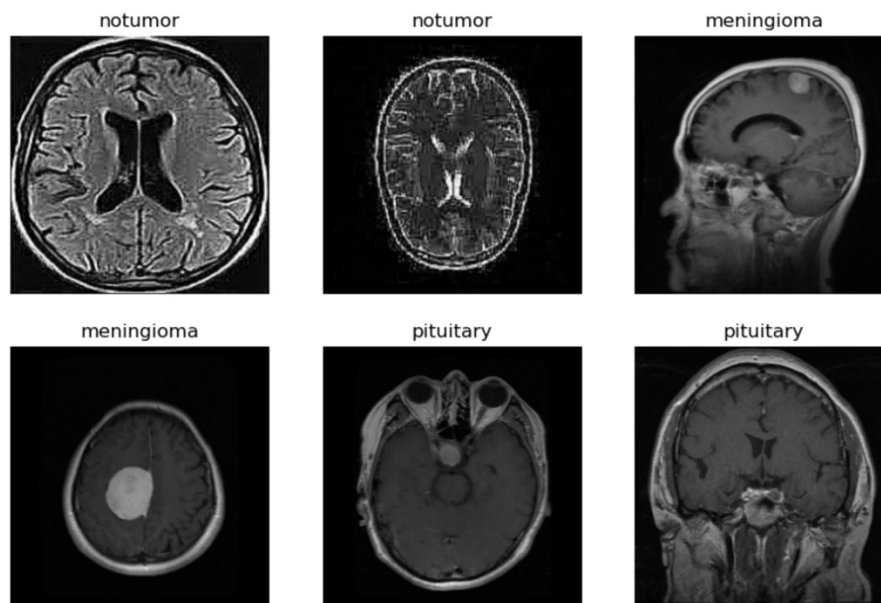


Figure 3.3.1: MRI scans from the two datasets

3.4 Data Augmentation

A machine learning approach which aims to expand the dataset size by applying specified translations and various transformations to the original data is referred to as data augmentation. With a larger dataset, the generalization abilities of the model can be improved substantially.

Since the dataset used in this study is small in size, applying data augmentation techniques will help expand the dataset size. It also helps reduce the chances of model overfitting. The data augmentation techniques applied in this study includes horizontal flip, vertical flip, 20% rotation and 0.2% zoom. All of these techniques were applied by creating pipelines of data using ImageDataGenerator available under the Keras library. ImageDataGenerator creates augmented images on the fly thereby greatly reducing memory utilization. The original and augmented versions generated by ImageDataGenerator for two dataset samples are shown in Figure 3.4.1. Moreover, the workflow used by ImageDataGenerator for augmentation is described in Figure 3.4.2. Firstly, a batch of images is selected from the dataset.

This is followed by image augmentation which is randomly applied to the batch of images. After this, the augmented image batch is sent for training.

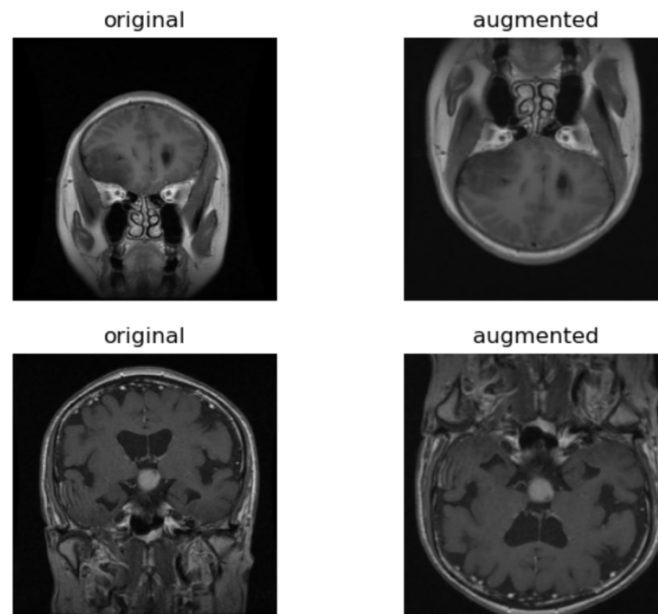


Figure 3.4.1: Original vs. augmented MRI generated by ImageDataGenerator

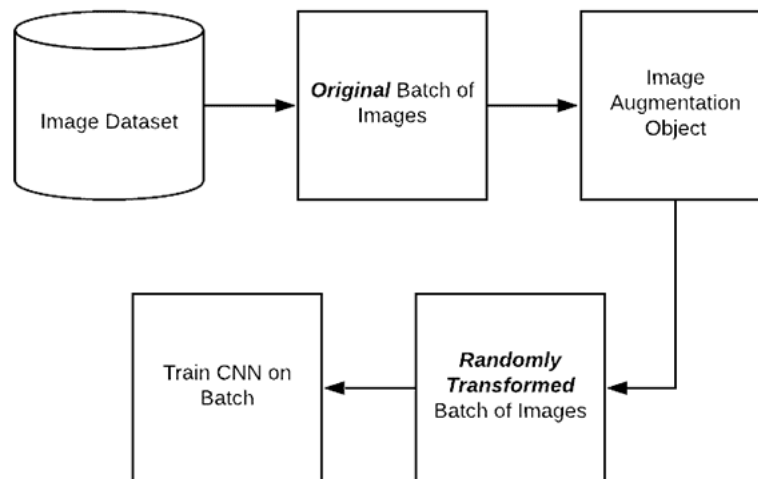


Figure 3.4.2: ImageDataGenerator workflow

3.5 Model with Transfer Learning

EfficientNetV2 is an improvement and extension of the previous EfficientNet (Tan & Le, EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks, 2020) models. EfficientNetV2 was presented by two Google

researchers in a research paper (Tan & Le, EfficientNetV2: Smaller Models and Faster Training, 2021). This architecture introduced notable improvements over the parent models. The compound scaling method was introduced to optimize the coefficients of the network's resolution, depth, and width. This greatly enhances the performance of the model without any impact on the size of the network. Moreover, EfficientNetV2 makes use of the Swish activation function in the place of ReLU which provides improved gradient flow and non-linearity in the model.

EfficientNetV2 comes in multiple variants. This study primarily focuses on experimenting with EfficientNetV2S model. Other variants include EfficientNetV2M and EfficientNetV2L. The key difference between these variants is the size of the network along with different input sizes. lists EfficientNetV2S's architecture chosen for the study. The minimum size of the acceptable input for EfficientNetV2S architecture is 128 x 128, whereas the maximum size is 300 x 300. Figure 3.5.1 demonstrates the architecture layers of the EfficientNetV2-S model.

Stage	Operator	Stride	Channels#	Layers#
0	Conv3x3	2	24	1
1	Fused-MBConv1,k3x3	1	24	2
2	Fused-MBConv4,k3x3	2	48	4
3	Fused-MBConv4,k3x3	2	64	4
4	MBConv4,k3x3,SE0.25	2	128	6
5	MBConv6,k3x3,SE0.25	1	160	9
6	FMBCConv6,k3x3,SE0.25	2	256	15
7	Conv1x1,Pooling,FC	-	1280	1

Figure 3.5.1: EfficientNetV2-S Architecture

3.5.1 MBConv

MBConv (Mobile Inverted Bottleneck Convolution) is the building block of all the EfficientNet models. It is made of three primary components: expansion, depth-wise convolution, and pointwise convolution.

- **Expansion Convolution:** An expansion factor is used in proliferating the input channels in this component. The bottleneck layer width is determined and controlled by this expansion factor. As the expansion factor increases, the capacity and computational cost of the model also increases.
- **Depthwise Convolution:** Convolution is performed on the input tensor with every filter operating on only a single input channel. This component minimizes the computational cost by avoiding convolution of the complete tensor at once.
- **Pointwise Convolution:** Once depthwise convolution is complete, pointwise convolution is applied. It uses a 1x1 filter to combine all the output of the depthwise convolution into a compact number of output channels. This component helps in minimizing the dimensionality.

In traditional use of MBConv, expansion factor is applied after the depthwise convolution. However, in an inverted MBConv expansion factor is applied before the depthwise convolution. The explained architecture of MBConv helps in lowering the computational cost due to the reduced count of parameters in the convolution stages which in turn balances the accuracy and efficiency of the model (Kavitha, Dhinakaran, Prabakaran, & Manigandan, 2024).

3.5.2 SE Operation

SE (Squeeze and Excitation) operation is used to improve the representational capability of CNNs. The SE operation was introduced in Squeeze and Excitation

network (SENet) models. It captures the channel-wise dependencies and readjusts the feature maps generated by the previous layers. A visual representation of this operation is provided in 3.5 The SE operation follows two steps:

- **Squeeze:** This step aggregates channel-wise information thereby reducing the spatial dimension. As shown in Figure 3.5.2, (where B = batch, H = height, W = weight, C = channels), global average pooling is applied across the spatial dimensions to achieve dimensionality reduction.
- **Excitation:** This step involves a fully connected CNN which is fed as input the squeezed feature maps from the preceding step. The result in this step is channel wise weights. Finally, these resulting weights are applied to the original feature map for recalculations. Figure 3.5.2 depicts the excitation operation with two fully connected layers, a layer with ReLU activation function, and a last layer with sigmoid activation function.

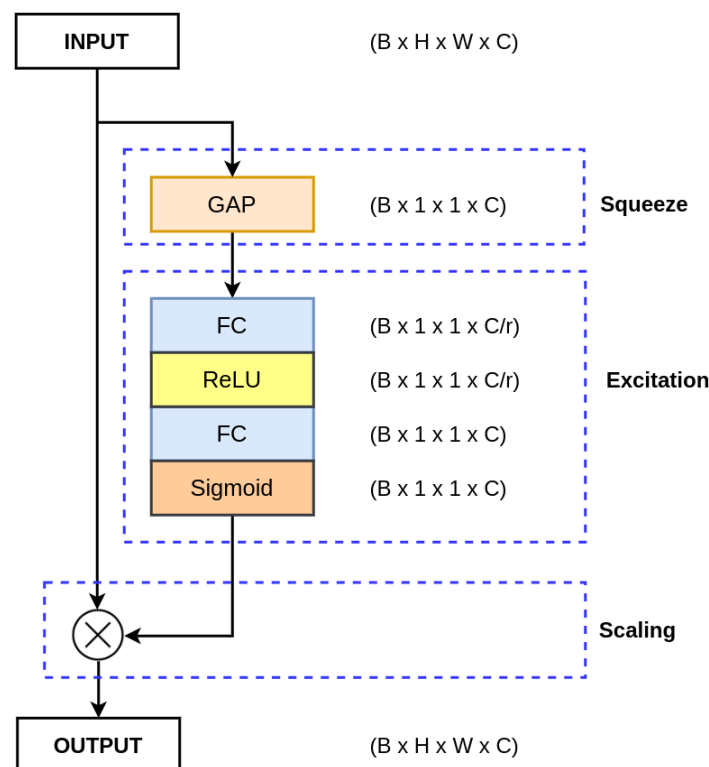


Figure 3.5.2: SE operation (Tan & Le, EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks, 2020)

3.5.3 Fused MBConv

FusedMBConv (Fused Mobile Inverted Bottleneck Convolution) is a variation of MBConv and is used for further optimization of the model's computational efficiency. This operator fuses multiple operations together thereby enhancing the computational efficiency of the MBConv operator (Kavitha, Dhinakaran, Prabakaran, & Manigandan, 2024). This further leads to a reduction in redundant operations and decrease in the frequency of memory access. The fused operations include depthwise convolution, expansion factor, pointwise convolution, and SE operation all combined in a single unit. As shown in Figure 3.5.3, the depthwise convolution in MBConv layer is merged with the final convolution layer in Fused-MBConv.

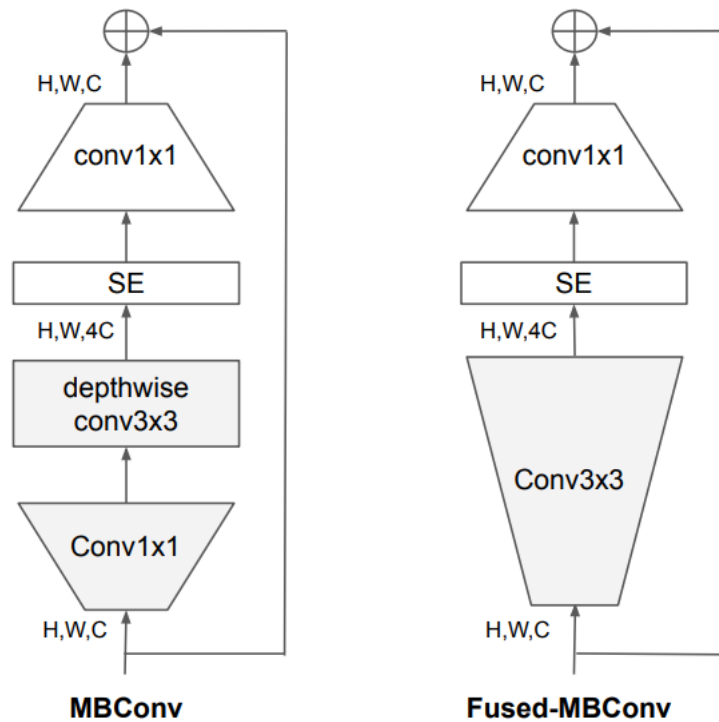


Figure 3.5.3: MBConv vs. FusedMBConv (Tan & Le, EfficientNetV2: Smaller Models and Faster Training, 2021)

3.6 Hyperparameter Selection

Hyper-parameters are crucial settings of the model that have a great effect on robustness, performance, generalization ability, and efficiency. Some of the most vital hyper-parameters of any CNN are activation function, count of hidden layers, dropout layer, and number of neurons (Koutsoukas, Monaghan, Li, & Huan, 2017). Other general hyper-parameters include batch size, learning rate, optimizer, epochs, loss function, etc. The hyper-parameters selected for this study are listed in Table 3.6.1.

Table 3.6.1: Selected hyperparameters

Hyperparameter	Value	Reasoning
Epochs:	50 (early stopping after 30)	Best value based on trial & error
Optimizer:	Adam	Best value based on trial & error
Batch size:	128	Best value based on trial & error
Learning rate:	1e-4	Best value based on trial & error
Loss function:	Categorical Cross entropy	Used for multiclass classification
Activation function:	SoftMax	Used for multiclass classification

3.7 Pooling Layers

In order to lower the dimensions of a newly generated feature map, pooling layers are applied. Two types of pooling layers are widely used, namely, max, and global average pooling. The most widely used pooling layer is the max pooling layer as it extracts the max value among each region. Global average pooling, however, computes the average value at each pooling region.

3.7.1 Max Pooling Layer

A technique frequently used for down sampling the spatial dimensions of the feature maps. A kernel slides across the feature map with a certain stride value. For every stride, the maximum value in the feature map's region is extracted and the left-over values are discarded. This technique aids in retaining the most

significant and prominent features from the feature maps. An example working of the max pooling layer is represented in Figure 3.7.1.

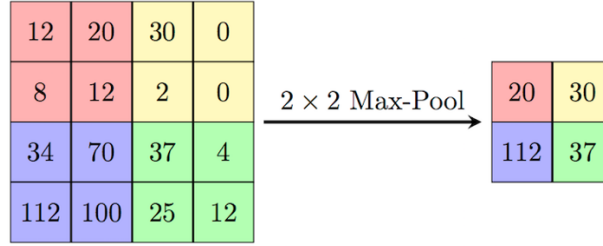


Figure 3.7.1: Max pooling layer representation

3.7.2 Global Average Pooling Layer

The spatial dimension of all the feature maps used can be significantly reduced by applying the Global Average Pooling 2D layer. The dimensions are reduced to a fixed size and value which is calculated by averaging the dimensions of the feature map. This allows to extract the paramount information and at the same time lowering the total number of parameters. Considering dimensions $W \times H \times K$, where W = width, H = height, and D = dimensions, and M as the feature map, then the formula for global average pooling 2D can be derived as shown in (3.1).

$$GAP(M) = \frac{1}{H * W} \sum_{i=1}^H \sum_{j=1}^W M(i, j) \quad (3.1)$$

3.8 Batch Normalization Layer

Batch normalization layer regularizes the activations of the layer before. Multiple batches are created during training. Batch normalization minuses the mean of every batch and divides it by the standard deviation of each of the batches to ensure that variance of the activations is always unit along with a zero mean. Introducing batch normalization layer increases the speed of training a network, prevents overfitting, and improves generalization.

3.9 Dense Layer

The dense layer is considered a fully connected layer in CNNs and is used in the final layers. In the dense layer, every previous neuron is interconnected to every neuron in the current layer with an associated weight. Furthermore, an activation function is attached to the network for non-linearity. Some commonly used activation functions consist of Softmax, ReLU, Tanh, Sigmoid, and Swish. Finally, the output layer of the CNN produces the final output providing the outcome of the task. Dense layer can be formulated as in the (3.2) where x = input layer, W = weight, b = bias, and z = output:

$$z = W \cdot x + b \quad (3.2)$$

3.10 Dense Layer with Softmax Activation – Output Layer

As discussed earlier, a fully connected layer with each neuron connected to every other neuron in the network is known as the dense layer. Moreover, the softmax activation function transforms the previous layer's output into a probability distribution over the different classes. The class with the highest probability is generally considered the result or predication made by the model.

3.11 Activation Function

Activation function is another important building block of a CNN. Its function is to regulate the activation of a neuron within a layer. Some of the common activation functions consists of: Sigmoid, Tanh, Rectified Linear Unit (ReLU), Parametric ReLU, Leaky ReLU, Swish etc.

Certain tasks and data require the use of specific activation functions. For instance, softmax activation function is used for multiclass classification tasks and swish activation function is considered a smooth function that provides improved gradient flow whilst training (Ramachandran, Zoph, & Le, 2017). Swish

activation function is used in EfficientNetV2S which can be computed as in (3.3) where x is the input and $\sigma(x)$ is sigmoid function of x .

$$\text{Swish}(x) = x \cdot \sigma(x) \quad (3.3)$$

3.12 Convolution

Convolution layers are the building blocks of CNNs. This layer performs the convolution operation on the input provided by the prior layer. During this operation, an odd-sized filter is applied to the input to generate a single output. The filter is then slid over by the set stride and padding values. The combined result of all the convolution operations generates a feature map that depicts the learned learning from the input, usually edges and curves.

3.13 Model Training

With the hyper-parameters selected, the model is run on the training dataset. The workflow of the entire proposed model is visually represented in Figure 3.13.1. The model trains on a k-fold cross validation model for where the value of k is 3 (chosen based on trial and error).

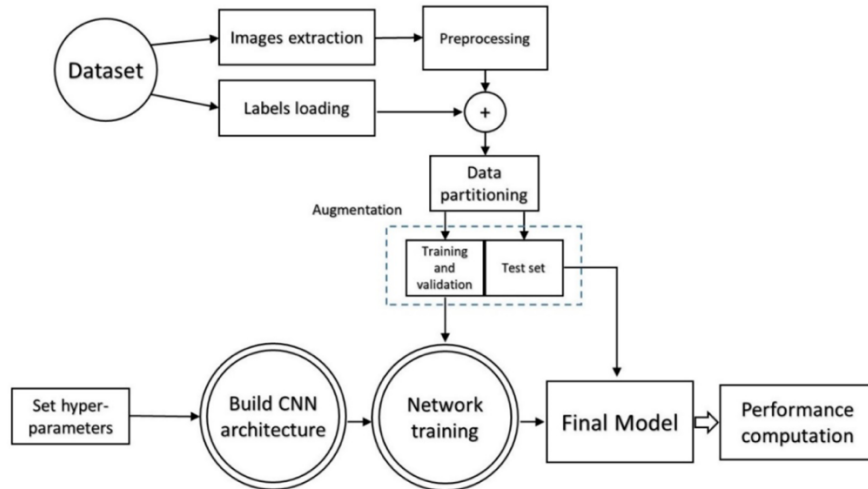


Figure 3.13.1: Visual diagram for the proposed model workflow (Incir & Bozkurt, 2024)

3.14 Model Optimization with Finetuning

After transfer learning is applied, fine-tuning could be utilized to investigate if further improvements can be made to the efficiency and robustness of the pre-trained model. This can be achieved by unfreezing several layers of the pre-trained model and retraining the network. Due to addition of new layers for training, the count of trainable parameters enlarges which improves the overall performance of the model. However, this can also raise the risk and chances of over-fitting.

3.15 Proposed Architecture and Experiments

EfficientNetV2S will serve as the base model for running the experiments. In this study, six experiments will be performed:

1. The first experiment will run with **no pooling layer** attached.
2. The second experiment will be conducted with a **global average pooling layer** attached.
3. The third experiment will be run by attaching the **max pooling layer** attached.
4. The fourth experiment will attach with the batch normalization layer **before** the dense layers.
5. The fifth experiment will run with the batch normalization layer **after** the dense layers.
6. Finally, the last experiment will **fine-tune** the best model from all the previous experiments to investigate if further improvements can be made to the performance of the model.

The proposed architecture for the top classifier for EfficientNetV2S includes:

- Pooling layer (*according to the experiment*)
- One flatten layer
- Two dense layers with varying units each
- One batch normalization layer (*according to the experiment*)

Chapter 4: Result

4.1 Introduction

This chapter discusses the results achieved by running the experiments detailed in Chapter 3. Various evaluation techniques were applied to assess the proficiency and correctness of the final CNN.

4.2 Metrics Used

To analyze the effectiveness and robustness of the proposed architecture, five metrics of evaluation were chosen, viz, precision, recall, f1-score, categorical accuracy, and AUC. Each of these selected metrics are discussed as follows:

- **Precision:** calculates the fraction of accurately predicted cases amongst all the positively predicted cases. Therefore, for precision metric, the focus lies on the accuracy of all the positive predictions made by the classifier. Precision can be calculated as in (4.1) where TP refers to the number of true positives and FP refers to the number of false positives.

$$Precision = \frac{TP}{TP + FP} \quad (4.1)$$

- **Recall:** computes the proportion of all the positively predicted occurrences to the overall number of accurate positive cases. A computed result of one represents the optimal recall, whereas the result zero signifies the bad recall. Recall can be calculated as in (4.2), where TP stands for the number of true positives, whereas FN stands for the number of false negatives.

$$Recall = \frac{TP}{TP + FN} \quad (4.2)$$

- **F1-Score:** combines the results of precision and recall yielding a balanced estimation of the algorithm's performance. F1-score can be calculated using (4.3) where precision and recall are defined in the points above.

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (4.3)$$

- **Categorical Accuracy:** used for multiclass classification tasks. This metric computes the number of rightly classified occurrences amongst all the samples available in the dataset. Categorical accuracy can be calculated using (4.4), where TP stands for the number of true positives, TN stands for the number of true negatives, FP stands for the number of false positives, and FN stands for the number of false negatives.

$$Categorical\ Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (4.4)$$

- **AUC:** computes the area under the ROC (Receiver Operating Characteristic) curve which is generated by plotting the false positive rate (FPR) against the true positive rate (TPR) at different thresholds for classification.

4.3 Model Evaluation

4.3.1 Experiment 1: No Pooling Layer

The model architecture utilized in this experiment is listed in Figure 4.3.1. Table 4.3.1 summarizes the results obtained by the model for the different metrics. It can be noted that training the model with no pooling layer garners a training loss of 0.0143. On the other hand, the testing losses for DS1 and DS2 are 0.1584 and 0.1020 respectively. The low values of both train and test loss suggest that the model is able to effectively predict the actual targets. Furthermore, it can be seen that the model was able to score 0.9979 in training f1-score, whereas the testing f1-scores for DS1 and DS2 were 0.9617 and 0.9609 respectively. It can be noted that the difference between the testing f1-scores for DS1 and DS2 is negligible even though DS2 is an unseen dataset. This concludes that the model has generalized well.

Layer (type)	Output Shape	Param	Trainable
input (InputLayer)	(None, 224, 224, 3)	0	Y
efficientnetv2-s	(None, 7, 7, 1280)	20331360	N
flatten (Flatten)	(None, 62720)	0	Y
dense (Dense)	(None, 128)	8028288	Y
output (Dense)	(None, 4)	516	Y
Total params: 28,360,164			
Trainable params: 8,028,804			
Non-trainable params: 20,331,360			

Figure 4.3.1: Experiment 1 architecture - no pooling layer

Table 4.3.1: Experiment 1 metrics - no pooling layer experiment

3-fold	Loss	Categorical Accuracy	Precision	Recall	F1-Score	AUC
Train	0.0143	0.9979	0.9979	0.9979	0.9979	1.0000
Val	0.0695	0.9785	0.9788	0.9783	0.9785	0.9984
Test DS1	0.1584	0.9610	0.9625	0.9610	0.9617	0.9825
Test DS2	0.1020	0.9697	0.9622	0.9697	0.9609	0.9846

The learning curve, shown in Figure 4.3.2, depicts an elbow curve which shows a gradual decrease in the training and validation loss before saturation. On the other hand, a gradual increase is visible in training and validation accuracy as well. The graph and metrics suggest the model to have a good fit.

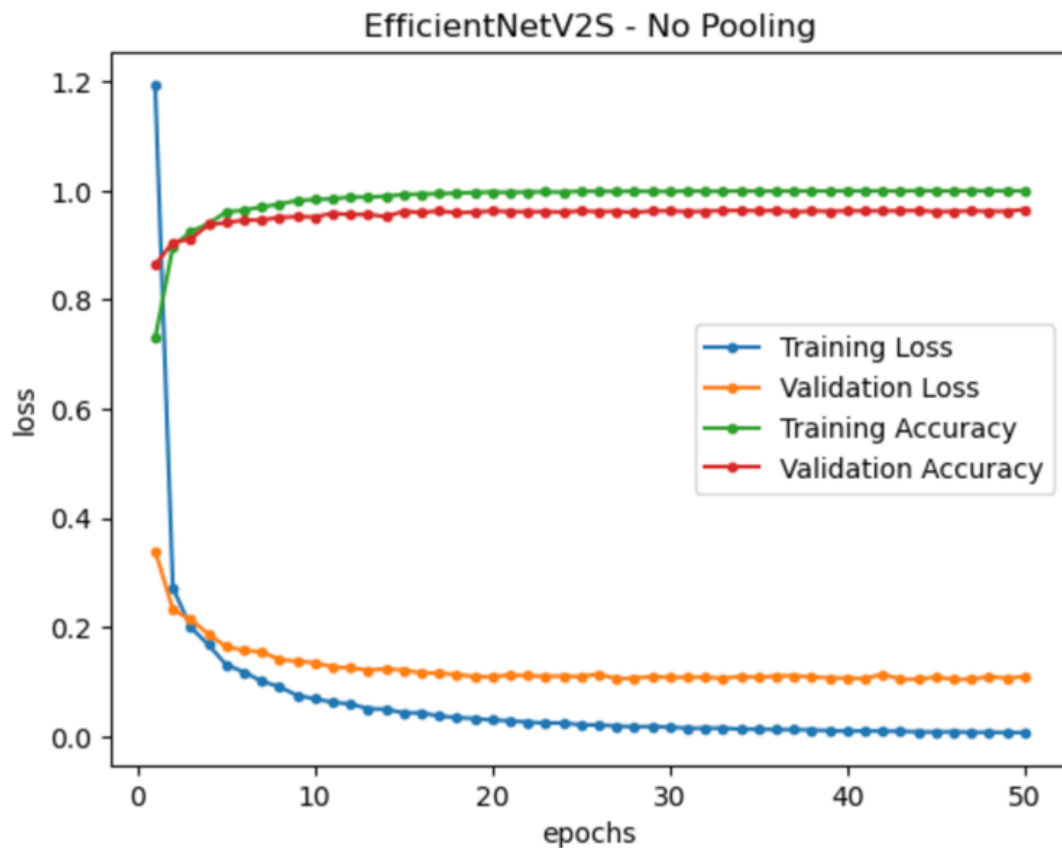


Figure 4.3.2: Experiment 1 - Loss Accuracy Curve - No Pooling

The confusion matrix for test datasets DS1 and DS2 are shown in Figure 4.3.3 and Figure 4.3.4 respectively. It can be noted that, for DS1, the best predicted class is pituitary, followed by glioma, no tumor, and meningioma. Similarly, for DS2, the best predicted class is pituitary, followed by, no tumor, glioma, and meningioma.

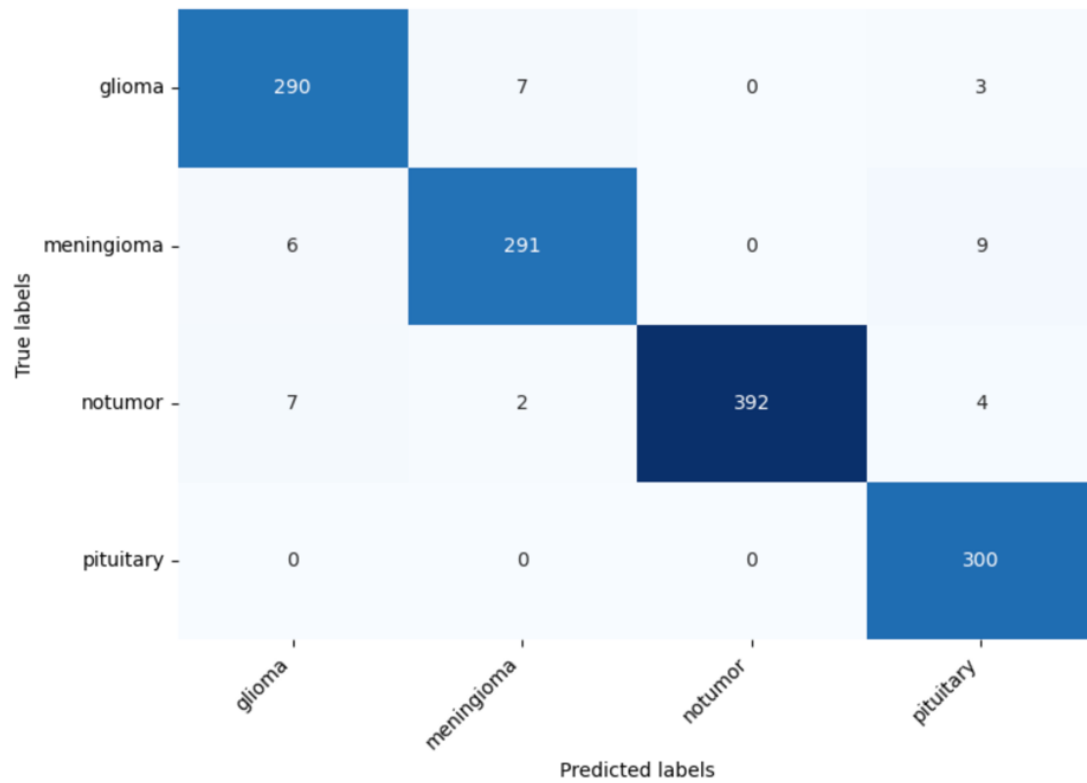


Figure 4.3.3: Experiment 1 - Confusion Matrix - No Pooling (Test DS1)

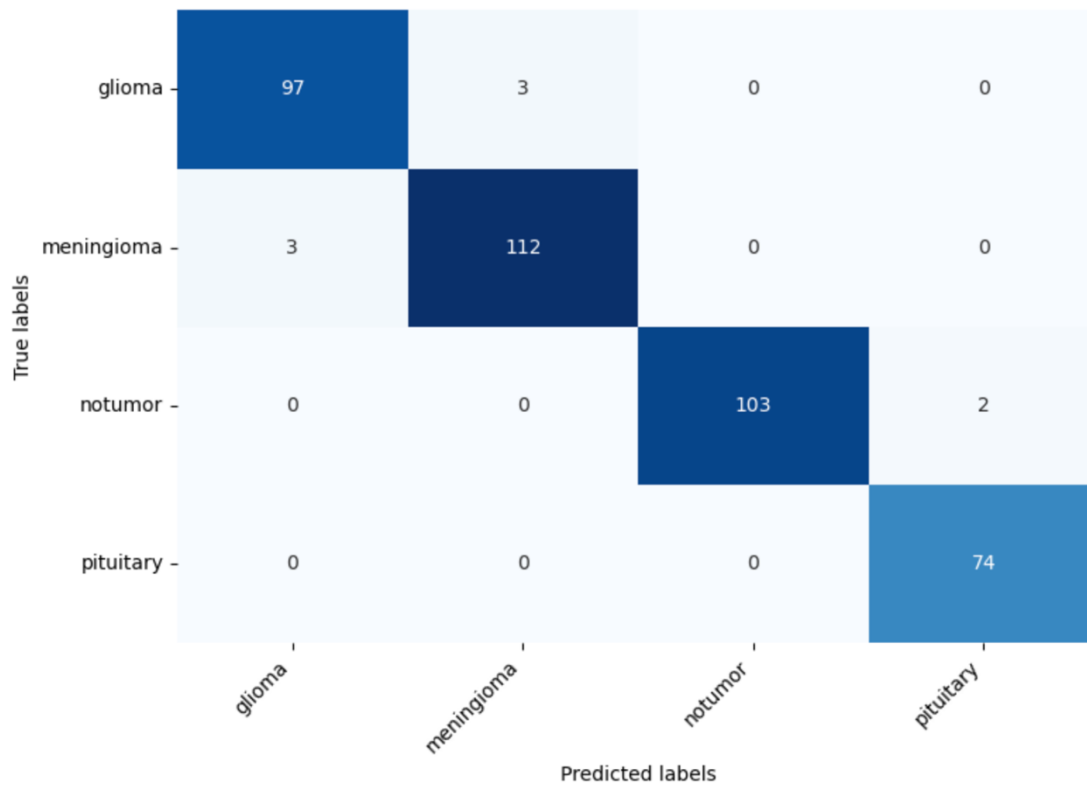


Figure 4.3.4: Experiment 1 - Confusion Matrix - No Pooling (Test DS2)

4.3.2 Experiment 2: Global Average Pooling Layer

The model architecture utilized in this experiment is listed in Figure 4.3.5. Table 4.3.2 summarizes the results obtained by the model for the different metrics. It can be noted that training the model with global average pooling layer garners a training loss of 0.2035. On the other hand, the testing losses for DS1 and DS2 are 0.2714 and 0.2992 respectively. These loss values are slightly higher when compared to the loss registered for Experiment 1. One reason for this could be the decrease in the number of trainable parameters in this experiment. Furthermore, it can be seen that the model was able to score 0.9291 in training f1-score, whereas the testing f1-scores for DS1 and DS2 were 0.9003 and 0.9032 respectively. It can be noted that the difference between the testing f1-scores for DS1 and DS2 is negligible even though DS2 is an unseen dataset. This concludes that the model has generalized well. However, these values are still lower in comparison to the results achieved by Experiment 1.

Layer (type)	Output Shape	Param	Trainable
input (InputLayer)	(None, 224, 224, 3)	0	Y
efficientnetv2-s	(None, 1280)	20331360	N
flatten (Flatten)	(None, 1280)	0	Y
dense (Dense)	(None, 128)	163968	Y
output (Dense)	(None, 4)	516	Y
Total params: 20,495,844			
Trainable params: 164,484			
Non-trainable params: 20,331,360			

Figure 4.3.5: Experiment 2 architecture - global average pooling layer

Table 4.3.2: Experiment 2 metrics - global average pooling layer experiment

3-fold	Loss	Categorical Accuracy	Precision	Recall	F1-Score	AUC
Train	0.2035	0.9286	0.9399	0.9185	0.9291	0.9936
Val	0.2415	0.9093	0.9193	0.8994	0.9092	0.9895
Test DS1	0.2714	0.8978	0.9123	0.8886	0.9003	0.9863
Test DS2	0.2992	0.9061	0.9186	0.8883	0.9032	0.9856

The learning curve, shown in Figure 4.3.6, depicts an elbow curve which shows a gradual decrease in the training and validation, however, no saturation is visible. On the other hand, a gradual increase is visible in training and validation accuracy as well. The graph and metrics suggest the model to have a good fit.

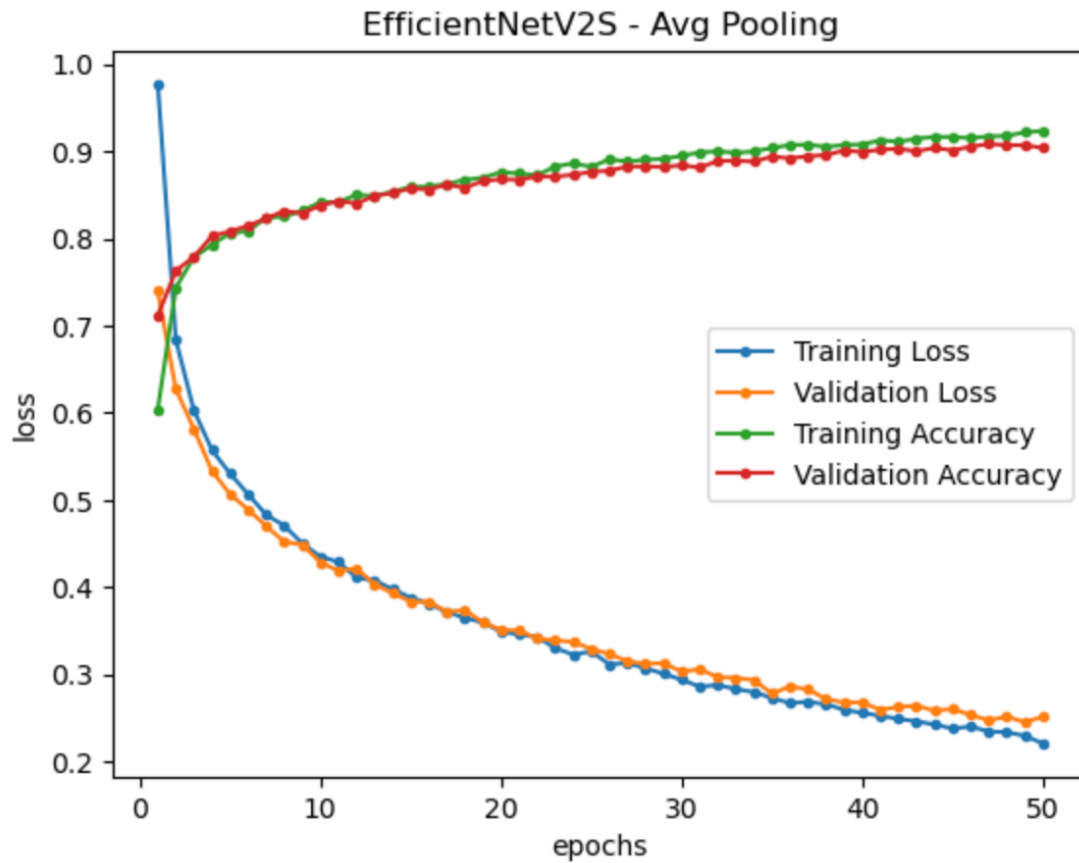


Figure 4.3.6: Experiment 2 - Loss accuracy curve - Global Average Pooling

The confusion matrix for test datasets DS1 and DS2 are shown in Figure 4.3.7 and Figure 4.3.8 respectively. It can be noted that, for DS1, the best predicted class is pituitary, following the no-tumor class, glioma, and meningioma. Similarly, for DS2, the best predicted class is pituitary, followed by, meningioma, no tumor, and glioma.

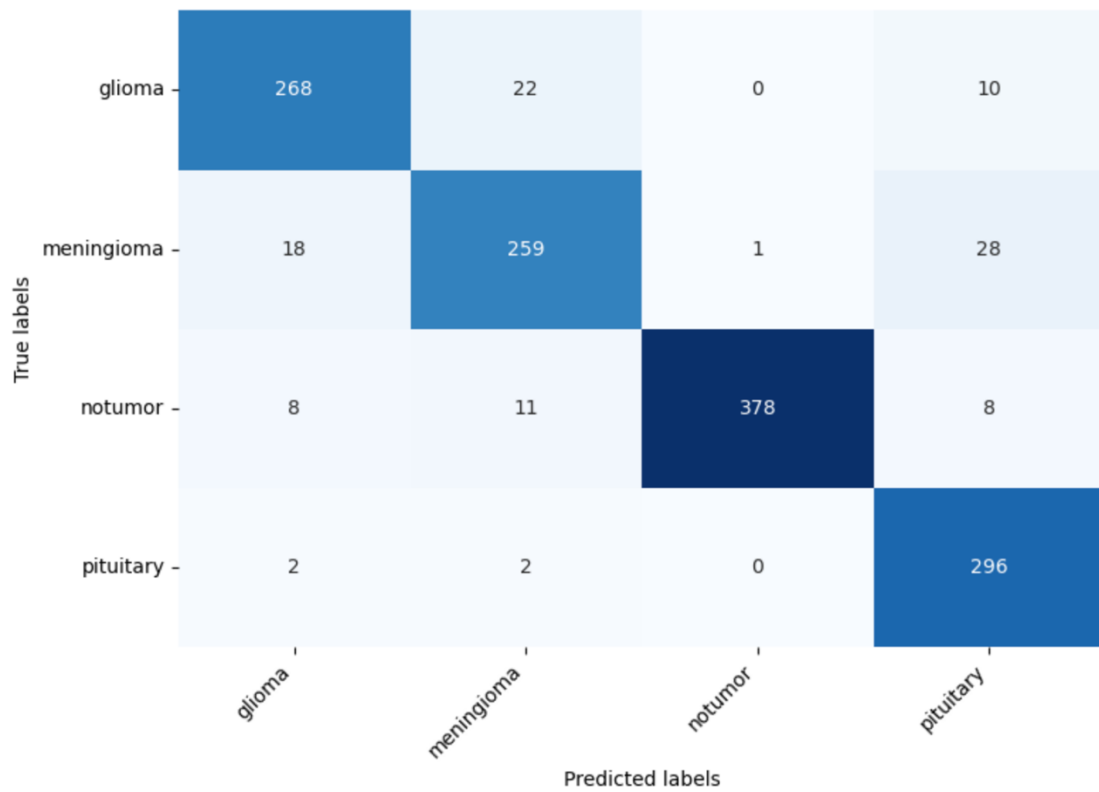


Figure 4.3.7: Experiment 2 - Confusion Matrix - Global Average Pooling (Test DS1)

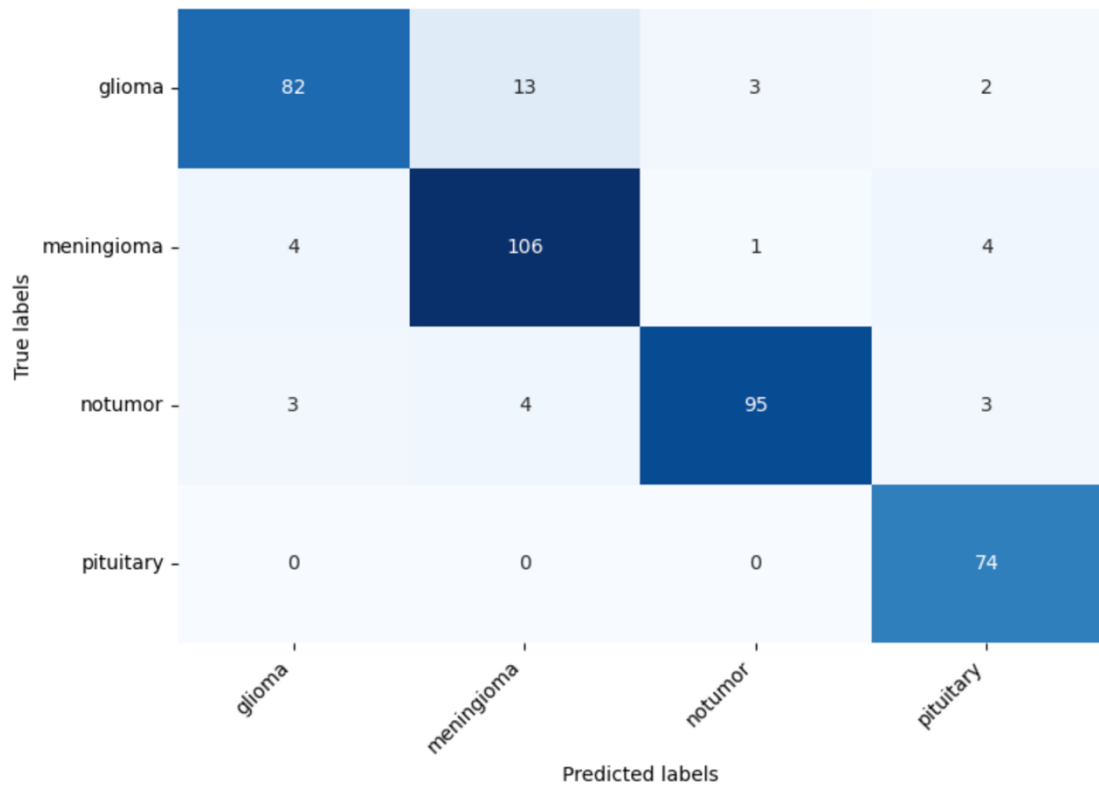


Figure 4.3.8: Experiment 2 - Confusion Matrix - Global Average Pooling (Test DS2)

4.3.3 Experiment 3: Max Pooling Layer

The model architecture utilized in this experiment is listed in Figure 4.3.9. Table 4.3.3 summarizes the results obtained by the model for the different metrics using max pooling layer. It can be noted that training the model with max pooling layer garners a training loss of 0.2634. On the other hand, the testing losses for DS1 and DS2 are 0.3309 and 0.5121 respectively. These loss values are significantly higher when compared to the loss registered for both Experiment 1 and 2. One reason for this could be the decrease in the number of trainable parameters in this experiment along with only the sharpest features being selected due to the max pooling layer. Furthermore, the model was able to score 0.9108 in training f1-score, whereas the testing f1-scores for DS1 and DS2 were 0.8796 and 0.8246 respectively. It can be noted that the difference between the training and testing f1-scores is significant which indicates that the model has not learned well from the training data.

Layer (type)	Output Shape	Param	Trainable
input (InputLayer)	(None, 224, 224, 3)	0	Y
efficientnetv2-s	(None, 1280)	20331360	N
flatten (Flatten)	(None, 1280)	0	Y
dense (Dense)	(None, 128)	163968	Y
output (Dense)	(None, 4)	516	Y
Total params: 20,495,844			
Trainable params: 164,484			
Non-trainable params: 20,331,360			

Figure 4.3.9: Experiment 3 architecture - max pooling layer

Table 4.3.3: Experiment 3 metrics - max pooling layer experiment

3-fold	Loss	Categorical Accuracy	Precision	Recall	F1-Score	AUC
Train	0.2634	0.9055	0.9169	0.9048	0.9108	0.9874
Val	0.3488	0.8682	0.8829	0.8533	0.8682	0.9775
Test DS1	0.3309	0.8810	0.8942	0.8654	0.8796	0.9799
Test DS2	0.5121	0.8096	0.8514	0.7995	0.8246	0.9553

The learning curve, shown in Figure 4.3.10, depicts an elbow curve which indicates gradual learning but also shows some irregularities in validation loss. There is also an increasing difference between the training and validation loss which suggests that the model has not learned well on the training data.

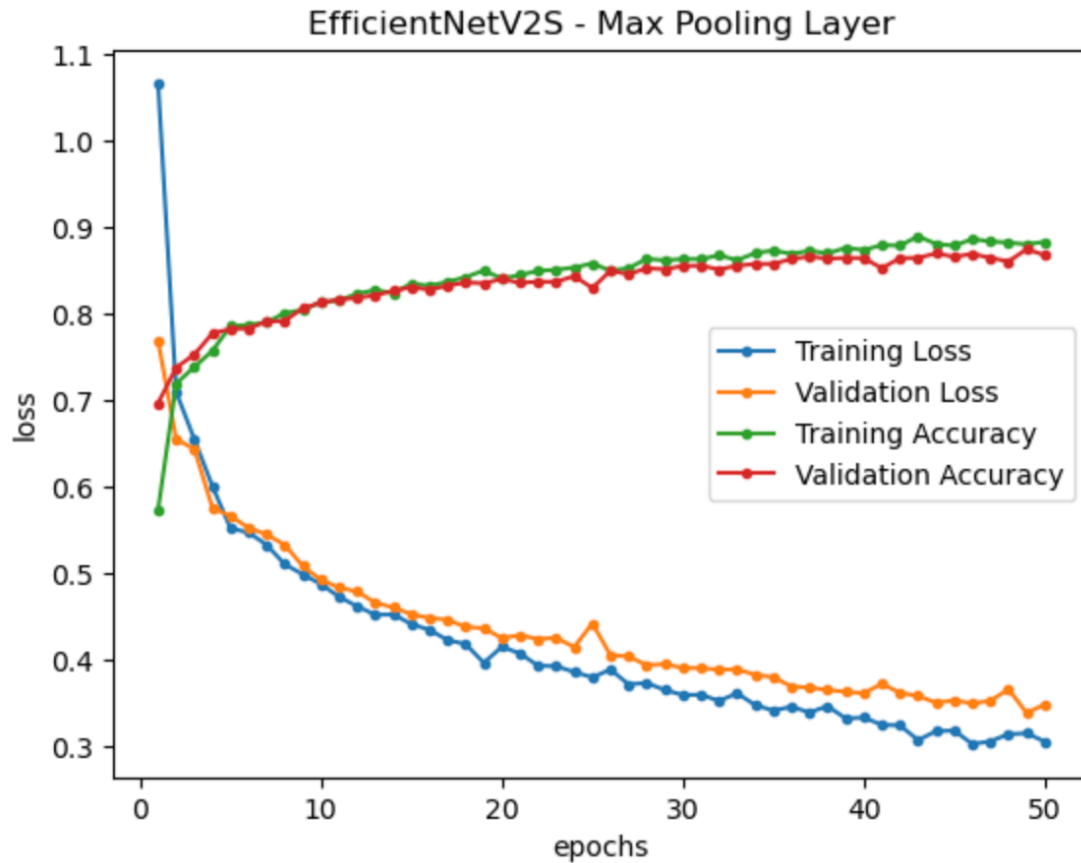


Figure 4.3.10: Experiment 3 - Loss accuracy curve - Max Pooling

The confusion matrix for test datasets DS1 and DS2 are shown in Figure 4.3.11 and Figure 4.3.12 respectively. It can be noted that, for DS1, the best predicted class is pituitary, following the no-tumor class, meningioma, and glioma. Similarly, for DS2, the best predicted class is pituitary, followed by no tumor, meningioma, and glioma. On the contrary, the worst predicted class for both DS1 and DS2 is glioma.

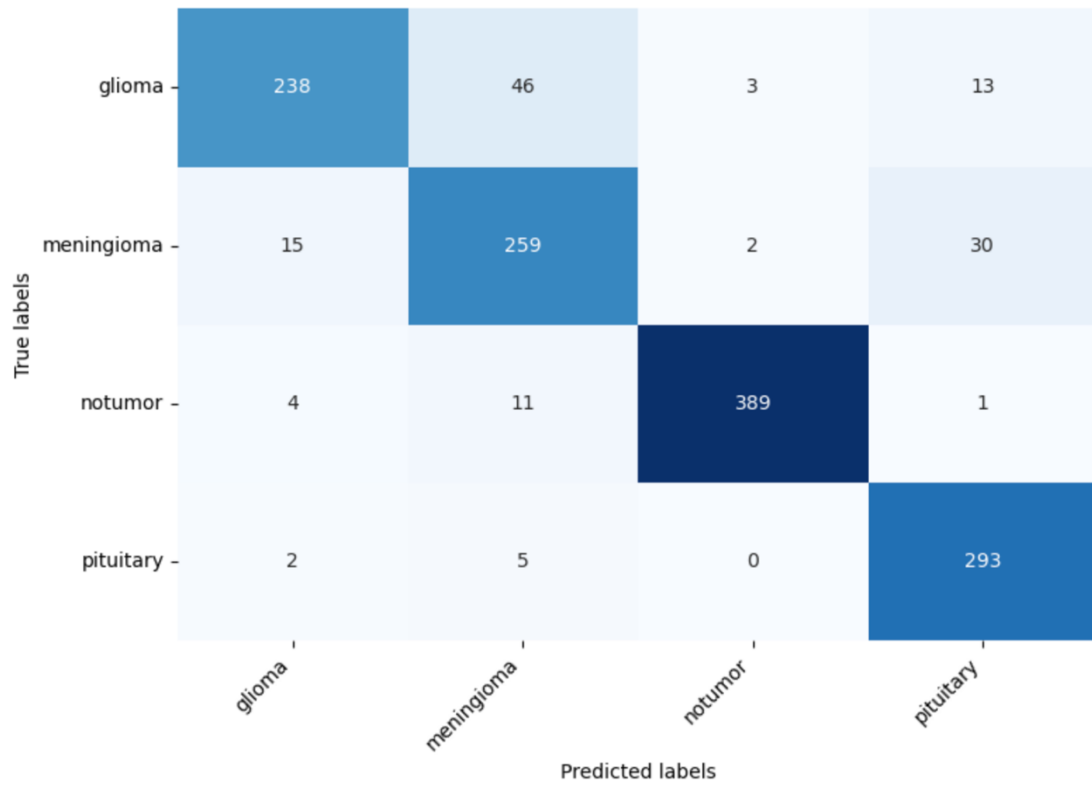


Figure 4.3.11: Experiment 3 - Confusion Matrix - Max Pooling (Test DS1)

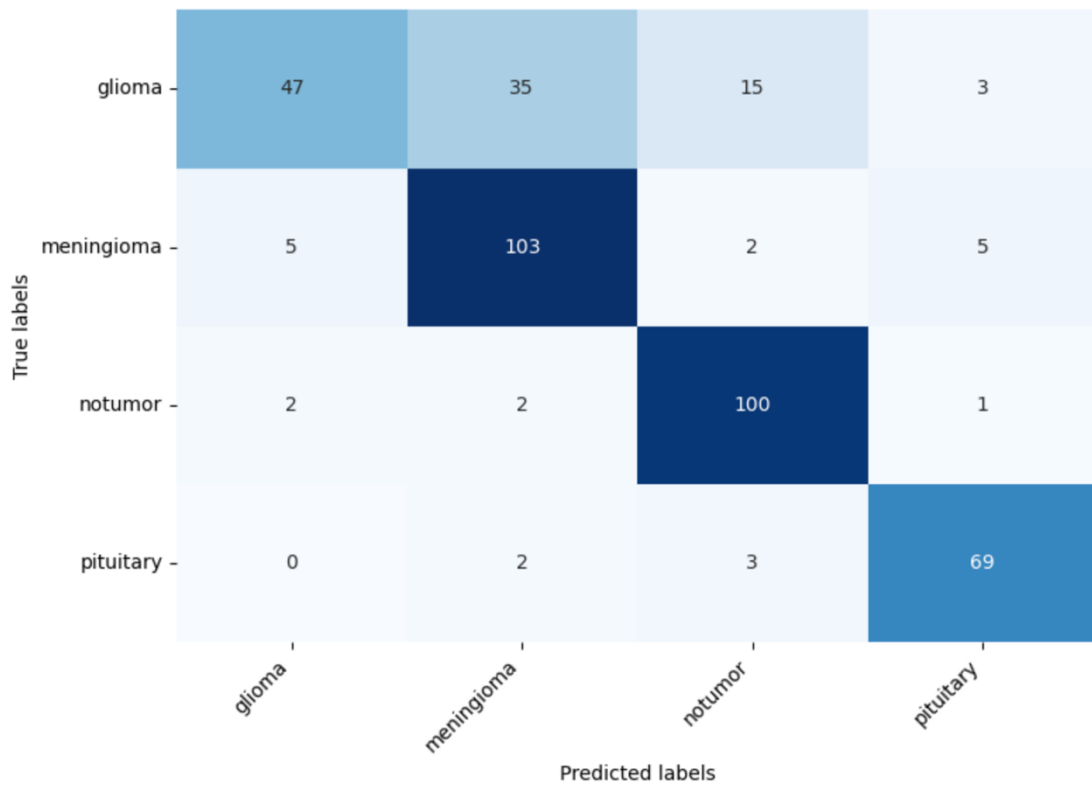


Figure 4.3.12: Experiment 3 - Confusion Matrix - Max Pooling (Test DS2)

4.3.4 Experiment 4: Batch Normalization before Dense Layer

The best model amongst experiments 1, 2, and 3 was selected for further experiments. In our case, experiment 1 (with no pooling) performed the best providing excellent evaluation metrics as discussed in Table 4.3.1. The model architecture utilized in this experiment is listed in Figure 4.3.13. Batch normalization layer is added to this experiment's model before the dense layer. Table 4.3.4 summarizes the results obtained by the model for the different metrics. It can be noted that the model garners a training loss of 0.0615 while, the testing losses for DS1 and DS2 are 0.1523 and 0.3037 respectively. Furthermore, the model was able to score 0.9804 in training f1-score, whereas the testing f1-scores for DS1 and DS2 were 0.9664 and 0.9645 respectively. It can be noted that there is a slight difference between the training and testing f1-scores which indicates that the model might not have learned well from the training data.

Layer (type)	Output Shape	Param	Trainable
input (InputLayer)	(None, 224, 224, 3)	0	Y
efficientnetv2-s	(None, 7, 7, 1280)	20331360	N
flatten (Flatten)	(None, 62720)	0	Y
batch normalization	(None, 62720)	250880	Y
dense (Dense)	(None, 128)	8028288	Y
output (Dense)	(None, 4)	516	Y
Total params: 28,611,044			
Trainable params: 8,154,244			
Non-trainable params: 20,456,800			

Figure 4.3.13: Experiment 4 architecture - batch normalization before dense layer

Table 4.3.4: Experiment 4 metrics - batch normalization before dense layer

3-fold	Loss	Categorical Accuracy	Precision	Recall	F1-Score	AUC
Train	0.0615	0.9804	0.9805	0.9804	0.9804	0.9986
Val	0.1715	0.9639	0.9646	0.9635	0.9642	0.9928
Test DS1	0.1523	0.9655	0.9663	0.9652	0.9664	0.9927
Test DS2	0.3037	0.9645	0.9645	0.9645	0.9645	0.9868

The learning curve, shown in Figure 4.3.14, depicts an elbow curve which indicates gradual learning but also shows significant irregularities in validation loss. There is also a constant difference between the training and validation loss which suggests that the model has not learned well on the training data.

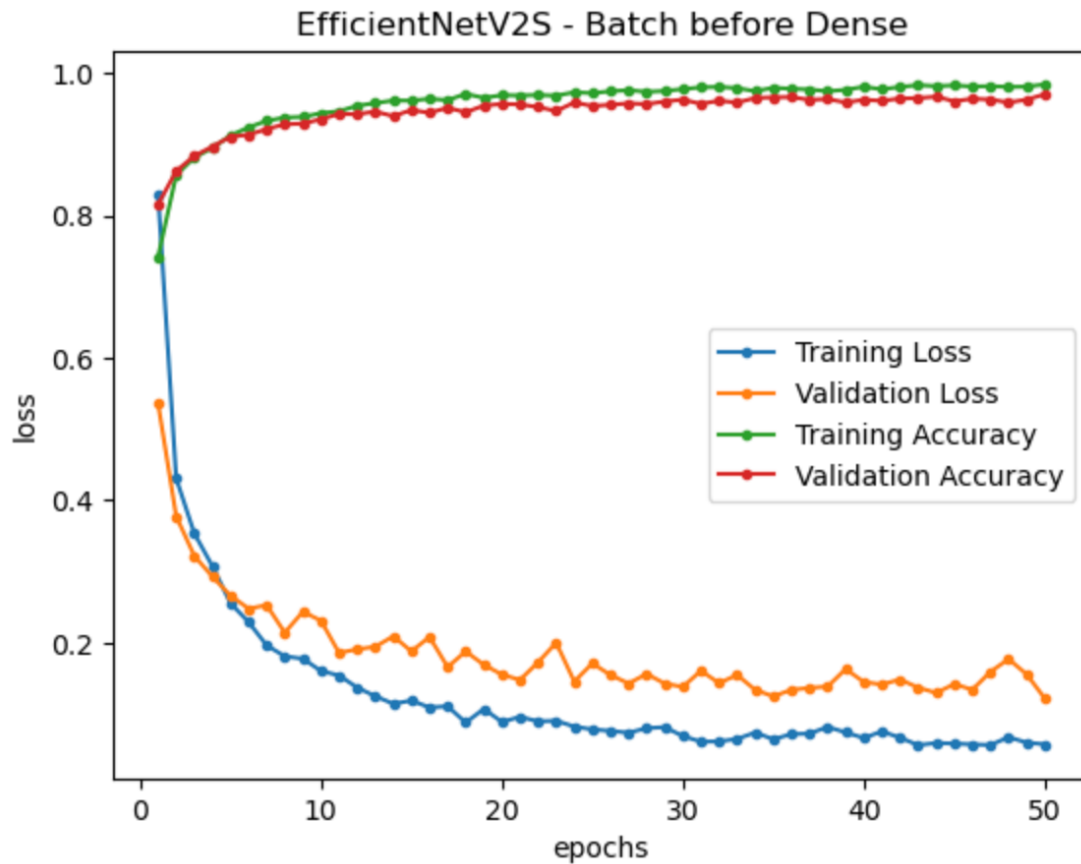


Figure 4.3.14: Experiment 4 - Loss accuracy curve - batch normalization before dense layer

The confusion matrix for test datasets DS1 and DS2 are shown in Figure 4.3.15 and Figure 4.3.16 respectively. It can be noted that, for DS1, the best predicted class is pituitary, followed by no-tumor class, meningioma, and glioma. Similarly, for DS2, the best predicted class is pituitary, followed by no tumor, meningioma, and glioma. On the contrary, the worst predicted class for both DS1 and DS2 is glioma.

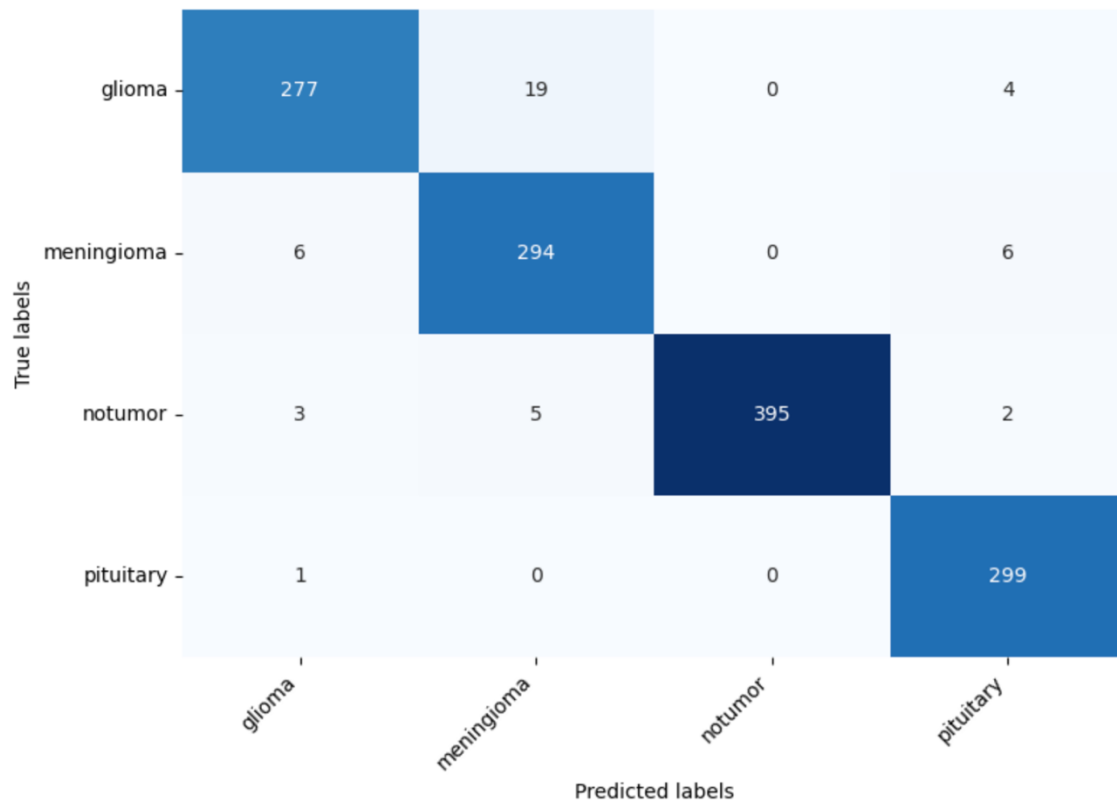


Figure 4.3.15: Experiment 4 - Confusion Matrix - batch normalization before dense layer (Test DS1)

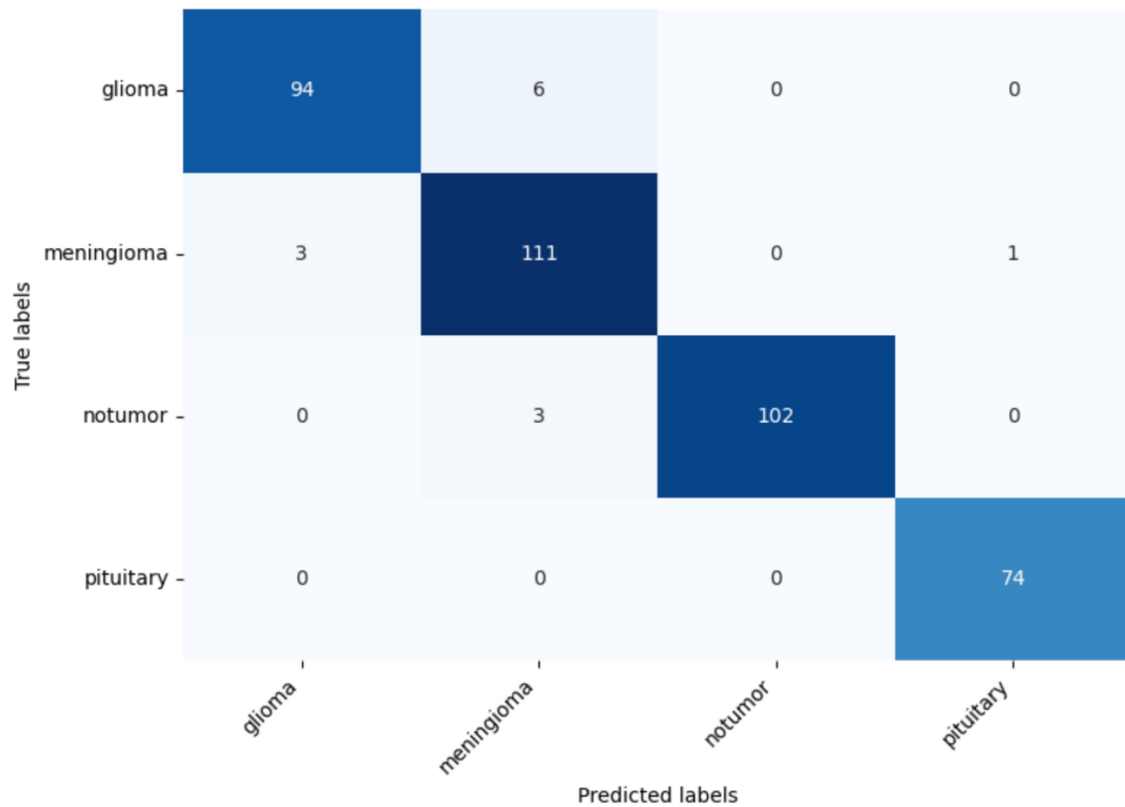


Figure 4.3.16: Experiment 4 - Confusion Matrix - batch normalization before dense layer (Test DS2)

4.3.5 Experiment 5: Batch Normalization after Dense Layer

The model architecture utilized in this experiment is listed in Figure 4.3.17. Table 4.3.5 summarizes the results obtained by the model for the different metrics. Batch normalization layer is added to this experiment's model after the dense layer. It can be noted the model garners a training loss of 0.0364. On the other hand, the testing losses for DS1 and DS2 are 0.1254 and 0.0983 respectively. Furthermore, the model was able to score 0.9836 in training f1-score, whereas the testing f1-scores for DS1 and DS2 were 0.9692 and 0.9798 respectively. This model performs better when compared to the results of experiment 4 where batch normalization layer was added before the dense layer.

Layer (type)	Output Shape	Param	Trainable
input (InputLayer)	(None, 224, 224, 3)	0	Y
efficientnetv2-s	(None, 7, 7, 1280)	20331360	N
flatten (Flatten)	(None, 62720)	0	Y
dense (Dense)	(None, 128)	8028288	Y
batch normalization	(None, 128)	512	Y
output (Dense)	(None, 4)	516	Y
Total params: 28,360,676			
Trainable params: 8,029,060			
Non-trainable params: 20,331,616			

Figure 4.3.17: Experiment 5 architecture - batch normalization after dense layer

Table 4.3.5: Experiment 5 metrics - batch normalization after dense layer

3-fold	Loss	Categorical Accuracy	Precision	Recall	F1-Score	AUC
Train	0.0364	0.9894	0.9903	0.9887	0.9836	0.9997
Val	0.0961	0.9702	0.9705	0.9702	0.9702	0.9945
Test DS1	0.1254	0.9687	0.9687	0.9687	0.9692	0.9922
Test DS2	0.0983	0.9797	0.9797	0.9797	0.9798	0.9945

The learning curve, shown in Figure 4.3.18, depicts an elbow curve which indicates gradual learning but also shows some irregularities in validation loss. There is also a constant difference between the training and validation loss which

suggests that the model has not learned well on the training data. However, this curve depicts smoother validation loss when compared to that of experiment 5 where batch normalization layer was added before the dense layer.

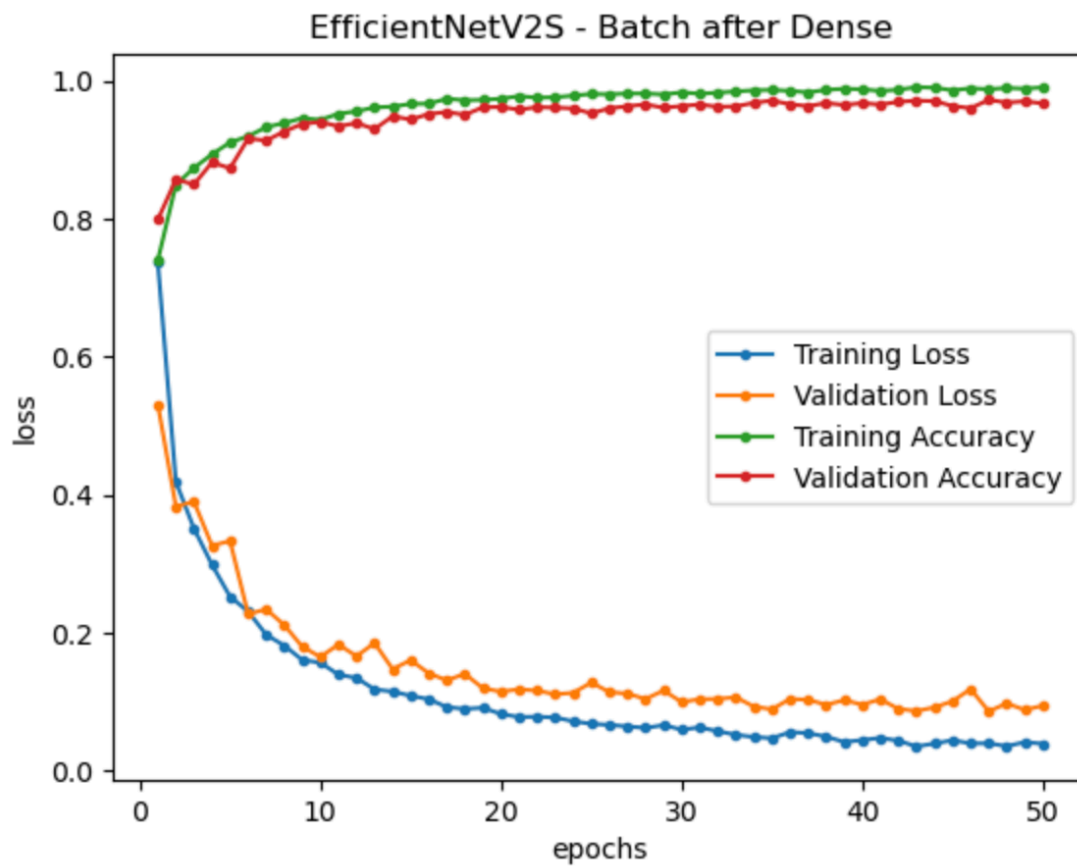


Figure 4.3.18: Experiment 5 - Loss accuracy curve - batch normalization after dense layer

The confusion matrix for test datasets DS1 and DS2 are shown in Figure 4.3.19 and Figure 4.3.20 respectively. It can be noted that, for DS1, the best predicted class is pituitary, followed by no-tumor class, glioma, and meningioma. Similarly, for DS2, the best predicted class is pituitary, followed by no tumor, meningioma, and glioma.

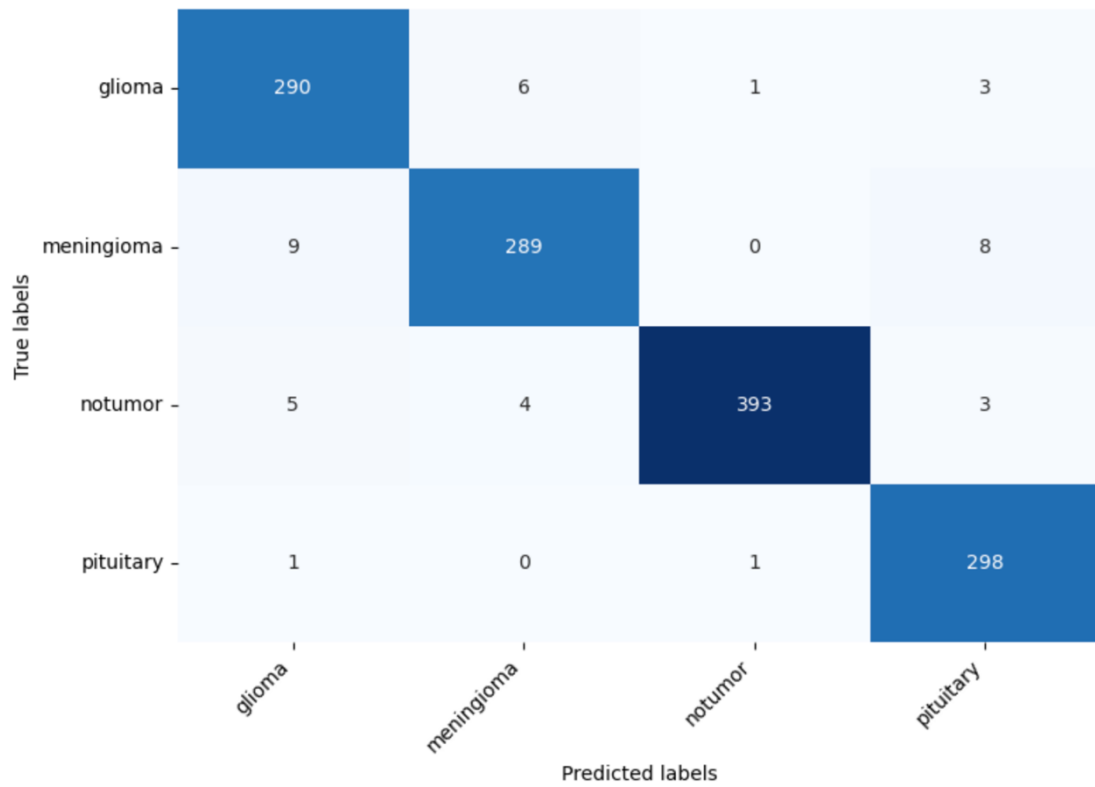


Figure 4.3.19: Experiment 5 - Confusion Matrix - batch normalization after dense layer (Test DS1)

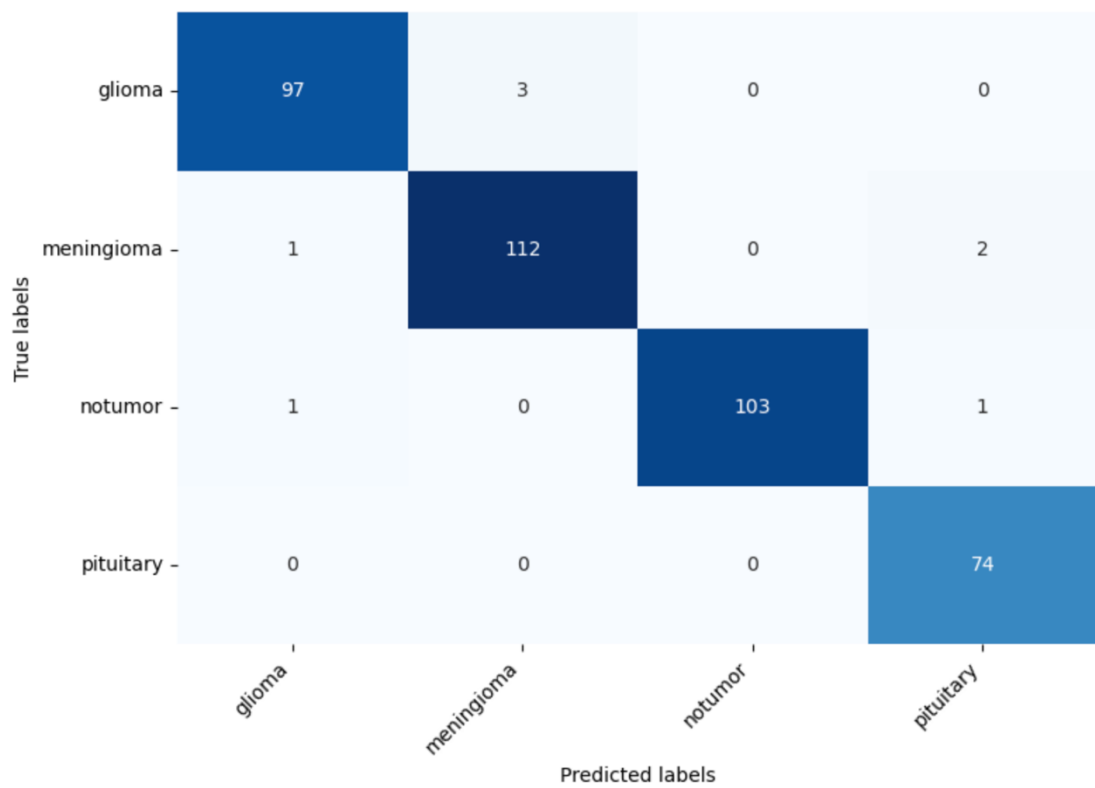


Figure 4.3.20: Experiment 5 - Confusion Matrix - batch normalization after dense layer (Test DS2)

4.3.6 Experiment 6: Finetuning the best classifier

The best model amongst all 5 experiments was selected for finetuning. In our case, experiment 5 (with no pooling and batch normalization after the dense layer) providing the best evaluation metrics as discussed in Table 4.3.5. The model architecture utilized in this experiment is listed in Figure 4.3.21. The last 36 layers of EfficientNetV2S were enabled for training after an extensive trial and error. Table 4.3.6 summarizes the results obtained by the model for the different metrics. It can be noted that the model garners a training loss of 0.0128 while, the testing losses for DS1 and DS2 are 0.0148 and 0.0232 respectively. Furthermore, the model was able to score 0.9962 in training f1-score, whereas the testing f1-scores for DS1 and DS2 were 0.9998 and 0.9947 respectively. It can be noted that the testing f1-scores recorded in this experiment are the highest values recorded among all the other experiments, signifying that this model performed the best out of all the other experimented models.

Layer (type)	Output Shape	Param	Trainable
input (InputLayer)	(None, 224, 224, 3)	0	Y
efficientnetv2-s	(None, 7, 7, 1280)	20331360	Y
flatten (Flatten)	(None, 62720)	0	Y
dense (Dense)	(None, 128)	8028288	Y
batch normalization	(None, 128)	512	Y
output (Dense)	(None, 4)	516	Y
Total params: 28,360,676			
Trainable params: 10,369,540			
Non-trainable params: 17,991,136			

Figure 4.3.21: Experiment 6 architecture - finetuning

Table 4.3.6: Experiment 6 metrics - finetuning

3-fold	Loss	Categorical Accuracy	Precision	Recall	F1-Score	AUC
Train	0.0128	0.9982	0.9980	0.9942	0.9962	1.0000
Val	0.0294	0.9903	0.9905	0.9901	0.9903	0.9996
Test DS1	0.0148	0.9998	0.9998	0.9998	0.9998	1.0000
Test DS2	0.0232	0.9947	0.9947	0.9947	0.9947	0.9998

The learning curve, shown in Figure 4.3.22, depicts an elbow curve which indicates gradual learning. There is almost no constant difference between the training and validation loss which suggests that the model has learned well on the training data and is able to predict the actual targets for every test.

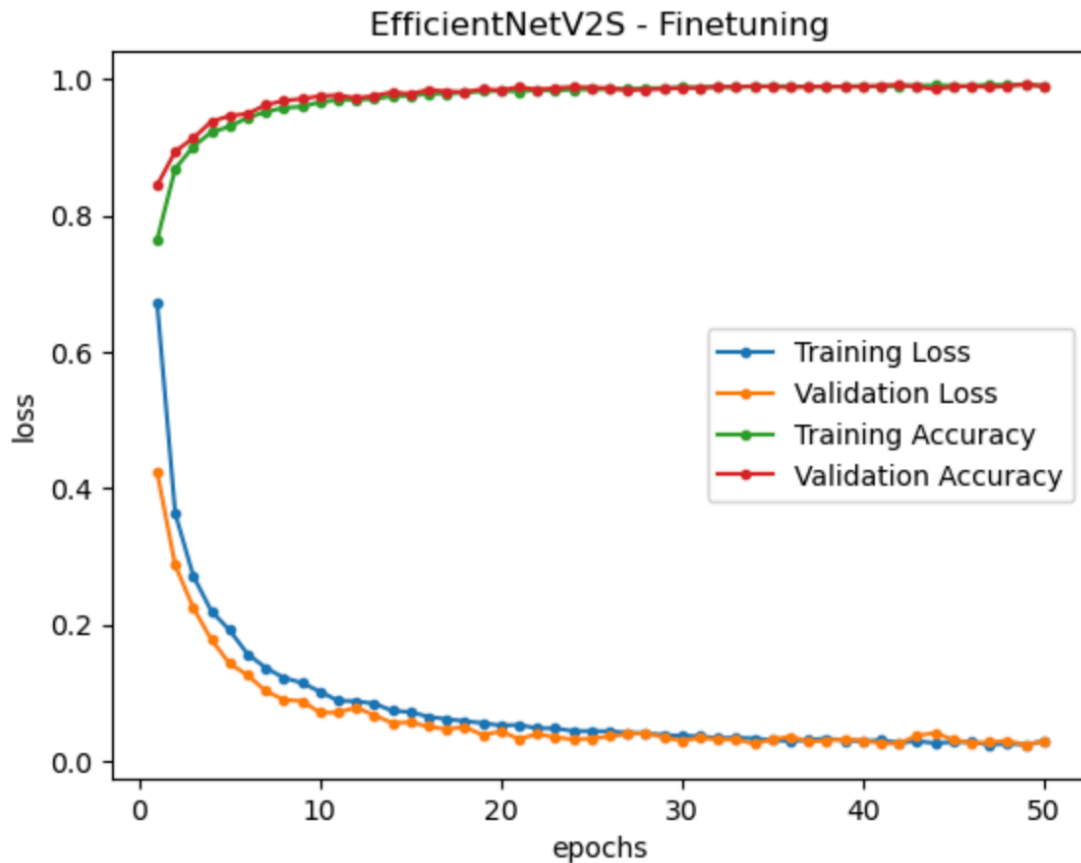


Figure 4.3.22: Experiment 6 – Loss accuracy curve - finetuning

The confusion matrix for test datasets DS1 and DS2 are shown in Figure 4.3.23 and Figure 4.3.24 respectively. It can be noted that, for DS1, the best predicted class is pituitary, followed by meningioma, no-tumor class, and glioma. Similarly, for DS2, the best predicted class is glioma, followed by pituitary, no tumor, and meningioma.

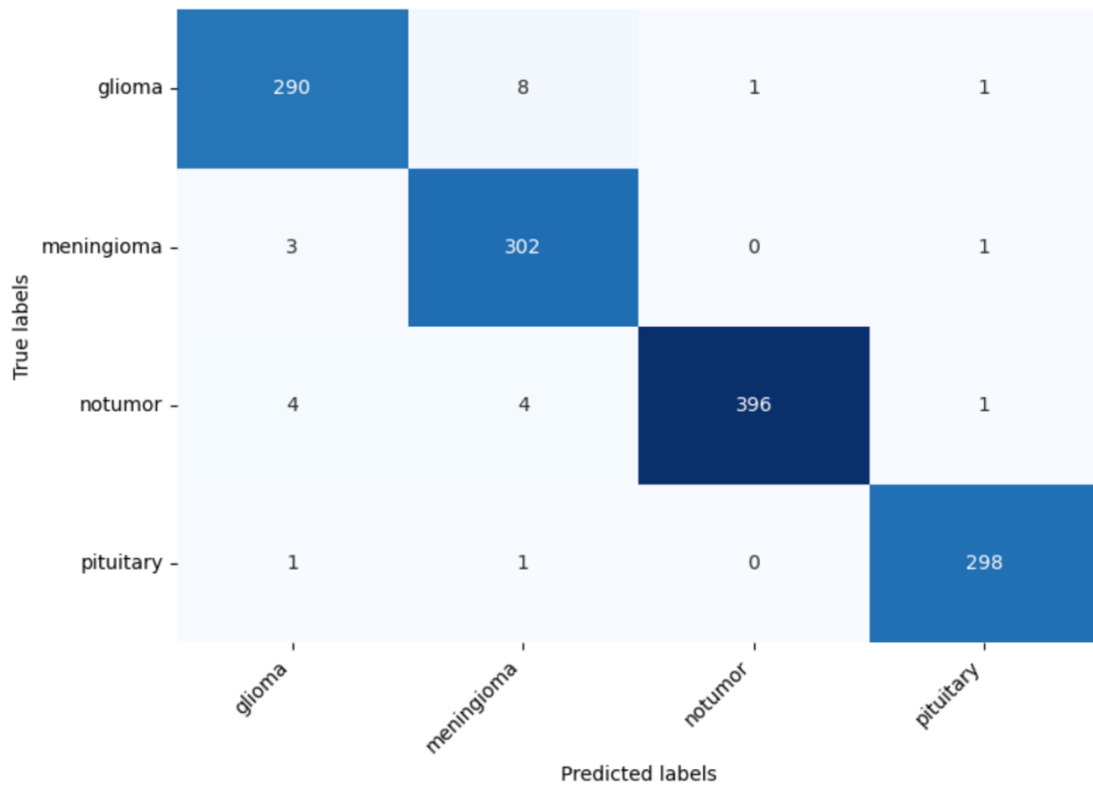


Figure 4.3.23: Experiment 6 - Confusion Matrix - finetuning (Test DS1)

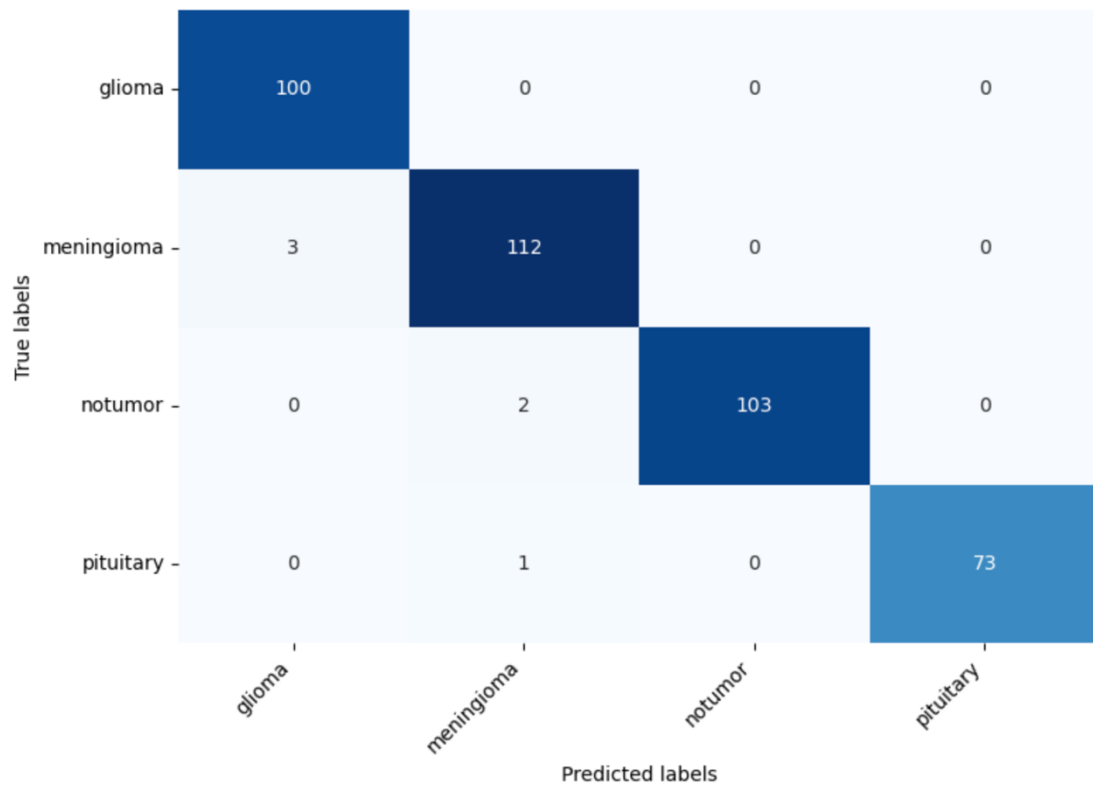


Figure 4.3.24: Experiment 6 - Confusion Matrix - finetuning (Test DS2)

4.4 Discussion

Through the results showcased in the previous section, it was concluded that the finest model was the fine-tuned model with no pooling layers and batch normalization layer added after the dense layer in the final classifier. It was noted that among no pooling layer, global average pooling layer, and max pooling layer, the best performing model was the one with no pooling layers added with f1-scores of 96.17% and 96.09% for the two test datasets. On the other hand, max pooling layer yielded the worst results with f1-scores of 87.96% and 82.46% for the two test datasets. Furthermore, the effect of placing batch normalization layer before and after the final dense layers was studied. It was determined that the model performed significantly better when the batch normalization layer was placed after the final dense layer garnering the f1-scores of 96.92% and 97.98% for the two test datasets. Finally, a fine-tuned model was proposed with no pooling layer and a batch normalization layer added after the final dense layer which resulted in f1-scores of 99.98% and 99.47% for the two test datasets, which is the highest results out of all the other experiments conducted. The proposed model was able to score 99.98% in training f1-score, which scoring 99.47% in testing f1-score. Finally, it can be derived that the placement and order in which layers are placed in the final classifier has an impact on the overall model performance.

Chapter 5: Conclusion, Limitations, and Future Work

5.1 Introduction

Investigating the application of pre-trained models for the brain tumor detection and classification task was the aim of this study. A pre-trained EfficientNetV2S model was investigated in this study. The layer placement and the order in which layers are added was also studied. This was done to explore if the model performance will experience any impact. It was found that the model without any pooling layer attached at the base model's end resulted in higher accuracy. Appending batch normalization layer **after** the dense layer showcased an increase in the accuracy when compared with batch normalization before the dense layer. Finally, the highest testing results were obtained from fine-tuning the model with no pooling layer and a batch normalization layer after the last dense layer in the final classifier.

5.2 Study Limitations

There are several limitations when working with brain tumor classification problem:

- **Limited availability of data:** Limited data is available in the field of brain tumor recognition. Most of the available datasets are built upon the existing .dicom/.nii dataset used for brain tumor segmentation problem. Most researchers have transformed the original dataset to work with PNG / JPG images instead of the original MRI scans. Therefore, it is difficult to test the models on new data.
- **Imbalance in classes:** Almost all the datasets used suffered from an imbalance resulting in poor classification of the smaller classes. This could

potentially lead to bias and result in lower accuracy when predicting the underrepresented classes.

- **Computational constraints:** The experiments conducted using deep learning models require powerful GPUs and hardware that can be computationally expensive. In another study, the access to enhanced computational power could yield better results or improved analysis.
- **Privacy and ethics:** Patient's security and data privacy needs to be maintained and therefore additional meta data cannot be acquired which can further help in identifying trends in the different patient demographics.

5.3 Conclusion

In conclusion, brain tumor classification is a significant area of study within the medicine. This study investigated the performance of EfficientNetV2S with different layers in the top final classifier. The aim of this study was to investigate if performance of the model can be enhanced by using, data augmentation, pooling layers, batch normalization layers, and finally by fine-tuning the most optimal performing model from the previous experiments. The most optimal model architecture for brain tumor classification in this study was the finetuned EfficientNetV2S with no pooling layer and batch normalization layer added after the dense layer. Not adding any pooling layer to the architecture provides better results when compared to addition of global average or max pooling layers. This could be because the key features (i.e., tumor) in brain tumor detection may not be the sharpest or the most dominant features. Therefore, these key features may be discarded when pooling layers are added since pooling layer only select the average or the sharpest features in the filter. Batch normalization layer added after the dense layer improves the performance of the model better than when added before the dense layer. This further suggests that the order in which layers are placed in the final classifier have an impact on the performance of the model.

Finetuning improved the performance of our model. Furthermore, the model was also tested on an unseen dataset that achieved almost similar results as the finetuned dataset, thereby confirming the generalization capability of the model and overruling any chances of overfitting.

5.4 Future Work

This study investigated EfficientNetV2S for brain tumor classification and showcased great results. Many other research avenues could be investigated to further improve the robustness and performance of the model. Following are the key areas proposed for future work within the domain:

- **Data Generation:** Currently, the datasets available are either small in size or face class imbalance issues. Research can be done of generating medical images with the help of Generative Adversarial Networks (GANs). This could help increase the size of medical datasets used for training and will eventually improve model generalization.
- **Different Imaging Modalities:** This study focused on the use of MRI scans for brain tumor detection. However, integration of different medical imaging modalities, such as, X-Ray, CT scan, or PET scan could be studied. This could help in building more diverse models.
- **Ensemble Learning:** Ensemble learning techniques could be explored in the future as stacking different models combines the strength and generates a more powerful model which could help enhance the precision and efficiency of the diagnosis.

References

- Alla, S., & Athota, K. (2022). Brain Tumor Detection Using Transfer Learning in Deep Learning. *Indian Journal of Science and Technology*, 2093-2102.
- Anaya Isaza, A., & Jiménez, L. M. (2022). Data Augmentation and Transfer Learning for Brain Tumor Detection in Magnetic Resonance Imaging. *IEEE Access*.
- Anaya Isaza, A., Mera Jiménez, L., Verdugo Alejo, L., & Sarasti, L. (2023). Optimizing MRI-based brain tumor classification and detection using AI: A comparative analysis of neural networks, transfer learning, data augmentation, and the cross-transformer network. *European Journal of Radiology Open*, 10, 100484.
- Bakas, S., Carr, C., Flanders, A., Kalpathy-Cramer, J., Mongan, J., Menze, B., & Prevedello, L. M. (2021). RSNA-MICCAI Brain Tumor Segmentation (BraTS) Challenge 2021. *International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI)*.
- Bhuvaji, S., Kadam, A., Bhumkar, P., Dedge, S., & Kanchan, S. (2020). Brain Tumor Classification (MRI). Kaggle.
- Brain Tumors*. (2024). Retrieved from American Association of Neurological Surgeons: <https://www.aans.org/en/Patients/Neurosurgical-Conditions-and-Treatments/Brain-Tumors>
- Chattopadhyay, A., & Maitra, M. (2022). Mri-based brain tumour image detection using cnn based deep learning method. *Neuroscience Informatics*.

- Dhakshnamurthy, V. K., Govindan, M., Sreerangan, K., Nagarajan, M. D., & Thomas, A. (2024). Brain tumor detection and classification using transfer learning models. *Engineering Proceedings*.
- Drevelegas, A. (2010). *Imaging of Brain Tumors with Histological Correlations*. Springer Berlin, Heidelberg.
- Goodenberger, M. L., & Jenkins, R. B. (2012). Genetics of adult glioma. *Cancer Genetics*, 205(12), 613–621.
- Incir, R., & Bozkurt, F. (2024). Improving brain tumor classification with combined convolutional neural networks and transfer learning. *Knowledge-Based Systems*, 299.
- Islam, M. M., Barua, P., Rahman, M., Ahammed, T., Akter, L., & Uddin, J. (2023). Transfer learning architectures with fine-tuning for brain tumor classification using magnetic resonance imaging. *Healthcare Analytics*, 4.
- Kavitha, P., Dhinakaran, D., Prabakaran, G., & Manigandan, M. (2024). Brain tumor detection for efficient adaptation and superior diagnostic precision by utilizing mbconv-finetuned-b0 and advanced deep learning. *International Journal of Intelligent Engineering and Systems*, 17(2).
- Koutsoukas, A., Monaghan, K. J., Li, X., & Huan, J. (2017). Deep-learning: investigating deep neural networks hyper-parameters and comparison of performance to shallow methods for modeling bioactivity data. *Journal of Cheminformatics*, 9(42).
- Kumar, K. A., Prasad, Y. A., & Metan, J. (2022). A hybrid deep cnn-cov-19-resnet transfer learning architype for an enhanced brain tumor detection and

- classification scheme in medical image processing. *Biomedical Signal Processing and Control*, 76(4).
- Litjens, G. (2017). A survey on deep learning in medical image analysis. *Medical Image Analysis*, 42, 60-88.
- Modiya, P., & Vahora, S. (2022). Brain Tumor Detection Using Transfer Learning with Dimensionality Reduction Method. *International Journal of Intelligent Systems and Applications in Engineering*.
- Osborn, A. G., Louis, D. N., Poussaint, T. Y., Linscott, L. L., & Salzman, K. L. (2022). The 2021 World Health Organization Classification of Tumors of the Central Nervous System: What Neuroradiologists Need to Know. *American Journal of Neuroradiology*.
- Ramachandran, P., Zoph, B., & Le, Q. V. (2017). Swish: A self-gated activation function. *Google Brain*.
- Rao, R. (2021). Brain tumor classification dataset.
- Shelatkar, T., Urvashi, Shorfuzzaman, M., Alsufyani, A., & Lakshmana, K. (2022). Diagnosis of Brain Tumor Using Light Weight Deep Learning Model with Fine-Tuning Approach. *Computational and Mathematical Methods in Medicine*.
- Srinivas, C., Prasad K. S., N., Zakariah, M., Alothaibi, Y., Shaukat, K., Partibane, B., & Awal, H. (2021). Deep Transfer Learning Approaches in Performance Analysis of Brain Tumor Classification Using MRI Images. *Journal of Healthcare Engineering*.
- Sultan, H. H., Salem, N. M., & Al-Atabany, W. (2019). Multi-classification of brain tumor images using deep neural network. *IEEE Access*, 69215-69225.

- Talukder, M., Islam, M., Uddin, M., Akhter, A., Pramanik, A., Aryal, S., . . . Moni, M. (2022). An efficient deep learning model to categorize brain tumor using reconstruction and fine-tuning. *Expert Systems with Applications*.
- Tan, M., & Le, Q. V. (2020). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *International Conference on Machine Learning*.
- Tan, M., & Le, Q. V. (2021). EfficientNetV2: Smaller Models and Faster Training. *International Conference on Machine Learning*.
- Tenghongsakul, K., Kanjanasurat, I., Archevapanich, T., Purahong, B., & Lasakul, A. (2023). Deep transfer learning for brain tumor detection based on mri images. *Journal of Physics: Conference Series*.
- Turner, & Nancy. (2022). *Cancer health concerns*.
- Wastika, A., Amda, J., Sihono, D., & Prasetyo, H. (2023). Efficientnetv2 based for mri brain tumor image classification. *International Conference on Computer, Control, Informatics and its Applications*, 171–176.

ملخص البحث

يعد التعرف الدقيق وفي الوقت المناسب على ورم المخ أمرا بالغ الأهمية لتخطيط العلاج لإطالة عمر المريض. فعلى مدى العقود الماضية، ركزت العديد من الجهود البحثية على تصنيف أورام الدماغ والكشف عنها. ومع ذلك، فمشكلة الورم الدماغي تعتبر من المشاكل البحثية الشاقة بسبب اختلاف الورم في الشكل والحجم والموقع. يعتبر التشخيص بالتصوير باستخدام الرنين المغناطيسي (MRI) اختبار تشخيصي شائع لتشخيص أورام الدماغ. تبحث هذه الدراسة في استخدام نهج التعلم العميق وباستخدام نموذج EfficientNetV2S، وهو نموذج مدرب مسبقا، لتشخيص أورام الدماغ. علاوة على ذلك، تمت دراسة موضع وترتيب إضافة الطبقات في المصنف النهائي العلوي وتم دمج زيادة البيانات لتوسيع حجم مجموعة البيانات. أخيرا، تم تطبيق الضبط الدقيق على النموذج الأفضل أداءات لتحسين أداء ودقة النموذج. تم تصنيف الأورام إلى أربع فئات: الورم الدبقي، والورم السحائي، والغدة النخامية، وعدم وجود ورم. تضمنت طبقة التصنيف النهائية طبقة كثيفة جنباً إلى جنب مع وظيفة تنشيط SoftMax. وخلصت الدراسة إلى أن النموذج الأعلى أداء هو نموذج EfficientNetV2S المضبوط بدقة مع عدم إضافة طبقة تجميع وطبقة تطبيع دفعة بعد الطبقة الكثيفة. تمكن النموذج المقترح من تسجيل 99.98٪ في تدريب f1، بينما سجل 99.47٪ في اختبار f1-score. وخلص أيضا إلى أن موضع الطبقات وترتيبها في المصنف النهائي لهما تأثير على أداء النموذج.

جامعة البحرين

كلية تقنية المعلومات



الكشف عن اورام المخ وتصنيفها باستخدام نهج التعلم العميق

أطروحة مقدمة كجزء من متطلبات الحصول على درجة الماجستير في تعلم الآلة والحوسبة الذكية

إعداد

مليحة مظفر علي ذو الفقار اسماعيل

202100104

إشراف

أ.د. علاء الدين يوسف العمري

(أستاذ)

جامعة البحرين

مملكة البحرين

ديسمبر 2024