CLASSIFICATION OF BRAIN TUMORS USING ARTIFICIAL INTELLIGENCE AND DEEP LEARNING

BY

CHAYUD MAHITHIPHARK 63011132

A PROJECT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF BACHELOR OF ENGINEERING IN BIOMEDICAL ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ACADEMIC YEAR 2023

SCHOOL OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

PROJECT CERTIFICATE

Project Title	Classification of brain tumors using artificial intelligence and deep learning
Student Name	Mr. Chayud Mahithiphark
	Student ID. 63011132
Degree	Bachelor of Biomedical Engineering
Project Advisor	Signed: _____
	(Dr. May Phu Paing)
Committee	Signed: _____
	(Assoc. Prof. Dr. Chuchart Pintavirooj)
Committee	Signed: _____
	(Assoc. Prof. Dr. Wibool Piyawattanamatha)
Committee	Signed: _____
	(Asst. Prof. Dr. Treesukon Treebupachatsakul)
Committee	Signed: _____
	(Asst. Prof. Dr. Kasama Srirussamee)
Head of Department	Signed: _____
	(Assoc. Prof. Dr. Sarinporn Visitsattapongse)

Project Title	Classification of brain tumors using artificial intelligence and deep learning
Student Name	Mr. Chayud Mahithiphark
Degree	Bachelor of Engineering in Biomedical Engineering
Project Advisor	Dr. May Phu Paing
Academic Years	2023

ABSTRACT

The advancement in Deep Learning (DL) has revolutionized medical diagnostics by enhancing the precision and automation of classification systems. In this project, we have developed a deep learning model for accurately categorizing brain tumors into three distinct types: meningioma, glioma, and pituitary tumor. Utilizing state-of-the-art architectures such as ResNet50V2, InceptionResNetV2, and DenseNet121 as our base models, we aimed to leverage their unique strengths in feature extraction and learning dynamics. To ensure the robustness and reliability of our model, we employed a five-fold cross-validation technique. This approach allowed us to test the model's performance across different data splits, ensuring consistency and generalizability. After that , we implemented the "model soup" technique, creating an ensemble of twenty individual models with varied hyperparameter combinations. This ensemble was evaluated using both uniform and greedy ensemble strategies to optimize performance. Our results indicated that the greedy soup approach closely matched the performance of the best individual model for each deep leatning architecture. For instance the best individual model using DenseNet121 and Greedy soup model of DenseNet121 have the same test accuracy of 98.7 %. In contrast, the Uniform soup model's test accuracy result is significantly lower than each individual model. Furthermore, we have initiated the integration of this DL model into a web application, aiming to establish it as a practical diagnostic tool and a platform for the efficient classification of brain tumors.

ACKNOWLEDGEMENTS

I want to express my appreciation to Dr. May Phu Paing, for her guidance, patience and expert advice throughout this research endeavor. Her insights and suggestions have played a role in shaping this work. Her encouragement has been a constant source of motivation.

I am also extremely grateful to all the professors who have contributed to my journey. Their unwavering commitment to excellence and high standards has truly. Enriched my work. Their willingness to share their knowledge and expertise has made an impact on my education.

Special thank to the researchers and participants who contributed to the collection of the dataset used in Cheng et al's study. Their diligent work, in compiling the data has played a role in advancing my research. The existence of such a dataset has proven to be an asset that greatly supports the analysis and findings presented in this study.

Lastly I extend my appreciation to my peers and colleagues for their companionship and thought provoking discussions that have greatly enhanced my thinking process.

Chayud Mahithiphark

TABLE OF CONTENTS

LIST OF TABLES

Tables Page

LIST OF FIGURES

Figures Page

LIST OF SYMBOLS/ABBREVIATIONS

Symbols/Abbreviations	Terms
AI	Artificial Intelligence
ML	Machine Learning
DL	Deep Learning
CNNs	Convolutional Neural Network
MRI	Magnetic Resonance Image
TP	True positives
FP	False positives
TN	True negatives
FN	False negatives
RGB	Red,green,blue color value
Relu	Rectified Linear Unit
JPEG	Joint Photographic Expert Group
GPU	Graphic Processing Unit
TPU	Tensor Processing Unit
CT	Computer Tomography

CHAPTER 1
INTRODUCTION

This chapter serves as an introduction to the themes discussed in this report and provides context for the work. It then outlines the reasons and objectives behind the project investigation followed by a summary of the project. Finally there is an overview of the dissertation highlighting each chapters content.

The field of engineering has seen a growing interest in artificial intelligence (AI) machine learning (ML) and deep learning (DL) due to their potential to transform healthcare. The rapid advancements in engineering and its impact on healthcare necessitate thorough investigations that utilize cutting edge technologies to address important research questions and drive technological progress. As biomedical engineering is still an emerging field there is plenty of room for methods and innovations that can make significant contributions to healthcare. The goal of this project is to contribute to the healthcare field by investigating and analyzing brain tumor classification using DL.

1.1 Background and significance of the study

In the world of biomedical engineering utilizing AI, ML and DL techniquesto medical diagnostics has become a significant focus area. The healthcare sector has acknowledged the potential of these technologies making it a captivating area, for investigation and technological advancements. [1] . The healthcare industry focus the potential of these technologies for exploration and technological progress.

One specific area where AI, ML and DL have shown promise is in the classification of brain tumors. Brain tumors encompass a group of neoplasms that can originate from different cell types and regions within the brain. Properly classifying brain tumors is crucial because it helps determine the suitable treatment approach. Different tumor types respond differently to treatments like surgery, radiation therapy or chemotherapy. However traditional methods of classifying brain tumors typically

involve histological analysis of biopsy samples, which can be subject to variability among observers.

Applying DL techniques to classify brain tumors offers an alternative to traditional methods. These approaches have the potential to provide accurate tumor classification using non invasive medical imaging data such, as magnetic resonance imaging (MRI) or computed tomography (CT) scans. Furthermore AI , ML and DL [2] algorithms can be trained to detect patterns in imaging data that might go unnoticed by human observers. This capability has the potential to aid in more precise diagnoses.

Despite the advancements made in this field there are still numerous challenges and unanswered questions that require further research. For instance the performance of DL algorithms may vary depending on factors like the quality and resolution of the input imaging data the features extracted from the data and the architecture of the algorithms themselves. Moreover there is a need for comprehensive validation studies to assess how well these approaches generalize across diverse patient populations and imaging techniques.

Therefore this study aims to tackle some of these challenges within brain tumor classification using DL techniques. The focus is on investigating and optimizing algorithms performance for accurately classifying brain tumors based on medical imaging data. Through this endeavor we hope to contribute knowledge to biomedical engineering while also providing valuable insights for developing more precise, reliable and clinically applicable tools, for brain tumor classification.

1.2 Objectives

The primary aim of this research project is to explore and analyze how advanced DL techniques can be used to classify brain tumors. The specific goal is to create a high performance classification model of identifying and categorizing brain tumors based on medical imaging data.

To achieve this objective we will first conduct a review of existing literature, on DL in the field of brain tumor classification. This literature review will provide us with an understanding of the state of the art methods, their limitations and the challenges that still need to be addressed in this area of research.

Following the literature review our focus will shift towards collecting a dataset of images for brain tumor classification. This dataset will include MRI and CT scans which will undergo processing techniques such as normalization, cropping, resizing and augmentation to make them suitable for DL applications. Once we have processed images at our disposal we will extract features using both handcrafted methods and DL approaches. These extracted features will then serve as input, for developing and training DL models.

The next step involves evaluating each model performance. Also including the implementing the "model_soups" method, which combines multiple models weight together as an ensemble to achieve the possible results, in classification. To evaluate the performance of each model we will use metrics such as accuracy, F1 score , Recall , sensitivity , and confusion matrix. We will also validate the results using cross independent datasets to ensure that the model can handle unseen data effectively.

1.3 Scope of the study

This study focuses on applying DL techniques for classifying brain tumors using medical imaging data. Our main focus will be on developing, training and evaluating each classification models weight. We will compare these weights to find the one using a method called "model_soups," which combines models together in an ensemble, for improved classification performance.

This study also aims to explore approaches, for extracting features from images in the context of DL. It will involve preprocessing the images to optimize their use in these applications. The performance of models will be evaluated using metrics and the results will be validated through cross validation and independent datasets. It's important to note that this study focuses on the aspects of classifying brain tumors using DL techniques. It does not delve into the evaluation or treatment of brain tumors. The study also solely relies on images as the source of data for classification without considering other data sources like genetics or patient history information. Additionally it should be acknowledged that the research is limited by the dataset of images which may restrict its scope in terms of including different types and variations of brain tumors.

1.4 Report outline

The rest of this report is organized as follows:

Chapter 2 of this report provides a literature review, on brain tumor classification using DL learning techniques. It offers an overview of state of the art methods, models and techniques used in this domain.This chapter also addresses the difficulties and gaps, in the existing literature setting the stage for the study.

Chapter 3 In this chapter we outline the methods and techniques utilized in this research. The chapter commences with an explanation of the data used including its source, nature and steps taken for pre processing. Subsequent sections delve into the design and implementation of the classification model providing insights into feature extraction, model architecture, training procedures and validation processes.

Chapter 4 demonstrates how Experimental Results and Discussion. This chapter presents the outcomes attained from conducting experiments as discussed in the chapter. It includes an analysis of classification performance through metrics, like accuracy, precision, recall, F1 score and Confusion Matrix. Furthermore it highlights the constraints faced throughout the research and explores directions, for investigations.

Chapter 5 closes the report, reviewing the work undertaken and draws conclusions about key parts of the work that was undertaken. Finally, future work is discussed with particular focus on conclusion and Future Work. This final chapter summarizes the main findings of the research and provides an overview of the contributions made by this study to the field of brain tumor classification. It also outlines the limitations encountered during the study and discusses the potential avenues for future research.

CHAPTER 2
REVIEW OF THEORY RELATED

This section delves into the techniques utilized in the classification of brain tumors along, with providing the background information for the design phase of this project. We will start by exploring types of brain tumors. The methods used to classify them in Section 2.1. Moving forward Section 2.2 will shed light on how MRI plays a role, in capturing high resolution images of the brain.

As we delve deeper into this study our focus shifts to DL models and techniques in Section 2.3. Here we will introduce Convolutional Neural Networks (CNNs) which're tools for image based tasks while also discussing the benefits of transfer learning. The sub sections that follow will provide details on architectures like ResNet50V2 and InceptionResNetV2.In Section 2.4 we will explore the significance of data preparation technique especially when it comes to imaging with MRI data. Additionally Section 2.5 emphasizes how the Data Augmentation in Medical Imaging can help in increasing size of dataset in datasets.Section 2.6 introduces a concept called Model SOUPs, which explores approaches like Uniform Soup and Greedy Soup. Lastly as we near the end of our discussion, in Section 2.7 we will delve into five fold cross validation which used to evaluate data preparation step. In Section 2.8 we analyse on how we are evaluating model performance using different matrix. Section 2.9 demonstrates how Streamlit serves as a connection making it easier to transfer our model onto a user web interface. Finally section 2.10 provides a summary of this chapter.

2.1 Brain Tumors and their Classification

Brain tumors encompass a range of conditions varying from non cancerous growths to malignant forms. They occur when cells, in the brain abnormally multiply potentially disrupting the functioning of this organ. The specific type, location and rate of growth of these tumors often determine the severity and variety of symptoms experienced by patients. While some tumors may not show any symptoms others can exert pressure on areas of the brain or disrupt pathways resulting in various neurological manifestations.

This study focuses on three types of brain tumors; glioma tumors, meningioma tumors, and pituitary tumors as shown in Figure 2.1. Gaining an understanding of these categories can provide insights into diagnosing, predicting outcomes and developing treatment strategies.

- Glioma Tumor; Gliomas are a group of tumors originating from cells that provide support and protection for neurons in the brain.Gliomas are further categorized into subtypes, like astrocytomas, oligodendrogliomas and ependymomas based on the type of cells involved.
- Meningioma Tumor; Meningiomas arise from meninges — layers of tissue that cover both the brain and spinal cord. Typically these growths, in the brain are slow growing and not cancerous. However their size and location can lead to symptoms.
- Pituitary Tumor; These growths occur in the gland a gland situated at the base of the brain that produces various hormones. Although most pituitary tumors are non cancerous they can disrupt hormone levels and cause symptoms. The pituitary gland, often referred to as the "master gland " is an organ, about the size of a pea that is located at the base of the brain. It has a role in regulating hormones that impact growth, metabolism, blood pressure and other important bodily functions. Tumors that develop in the gland although mostly non cancerous can result in a range of complications. These tumors may release hormones leading to conditions such as acromegaly or Cushings syndrome. Alternatively they can put pressure on structures, like the nerve and cause vision issues.[3]

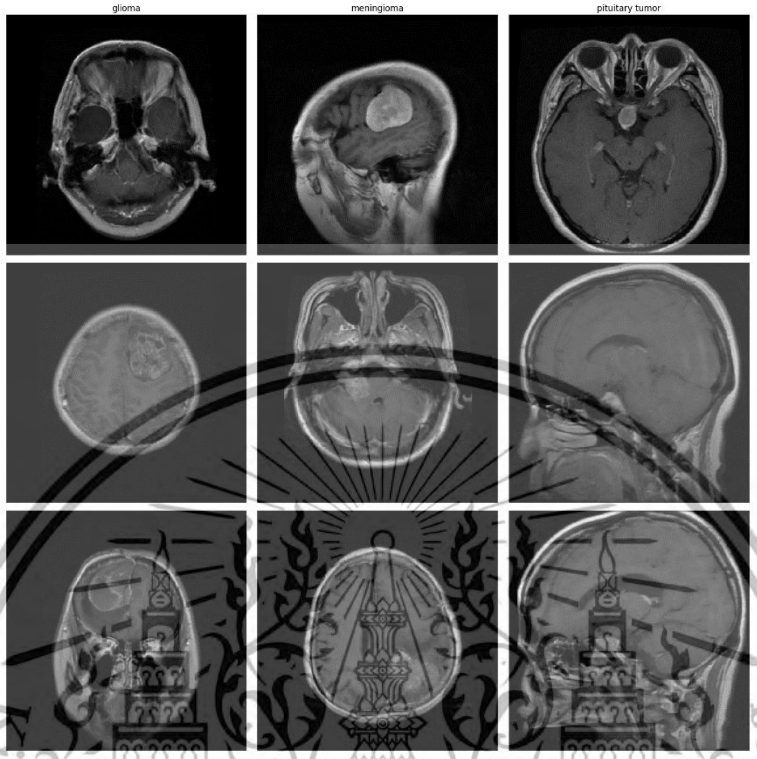

Figure 2.1 Sample in each classe of brain tumors

2.2 Magnetic Resonance Imaging

MRI is a tool, in the field of imaging. It combines fields, radio waves and advanced computer technology to create images of the bodys internal structures. [3]Unlike X rays and Computer Tomography (CT) scans MRI doesn't use radiation making it a safer option. This makes it particularly useful for examining tissues like muscles, tendons and the intricate structures of the brain. MRI ability to detect differences in tissues is crucial for identifying conditions such, as brain tumors. These tumors can be difficult to detect in their stages. MRI can accurately capture even the most nuanced tissue variations. Additionally MRI goes beyond two images by producing comprehensive three dimensional data through a series of contiguous slices.

Having an multifaceted understanding is crucial when studying the intricate nature of the brain. This becomes especially important when using techniques, like

MRI, which can offer a perspective and identify abnormalities. To enhance the effectiveness of MRI scans contrast agents such as gadolinium are used. These agents help highlight areas or structures of interest making it easier to visualize and accurately diagnose conditions, like brain tumors.[3] By accentuating the boundaries of tumors these contrast agents enable distinctions. Improve diagnostic precision.

2.3 Machine Learning Models and Feature Extraction Techniques

In this section, we establish the context and significance of various classification methods used in brain tumor classification.

2.3.1 Convolutional Neural Networks

CNNs have brought about a transformation, in the field of image processing and computer vision [1]. These networks are specifically designed to handle grid data, such as images and have emerged as a force in tasks like image classification, including the important area of classifying brain tumors. Layers in CNNs include such as pooling layers and connected layers. The role of layers is to apply filters over the input data and generate a feature map that captures the characteristics present within the image. Pooling layers then reduce the dimensions of this feature map simplifying information and making computations more efficient. Finally connected layers take these features and produce the ultimate classification outcome.

CNNs are highly effective when it comes to classifying brain tumors. One of their advantages is that they can analyze images without requiring manual extraction of features in image which might lead to overlook in importtant information during the extraction process. Additionally ,CNNs architectures are trained on datasets enabling them to learn and recognize a range of features that can be applied to new and unseen

data. This ability to generalize makes CNNs particularly well suited for classifying brain tumors as the appearance of tumors can vary greatly between patients.

To summarize CNNs have become a tool in the field of brain tumor classification because they automatically learn spatial features from medical images. By managing dimensional images and generalizing across diverse data CNNs have demonstrated remarkable accuracy, in detecting and classifying brain tumors.

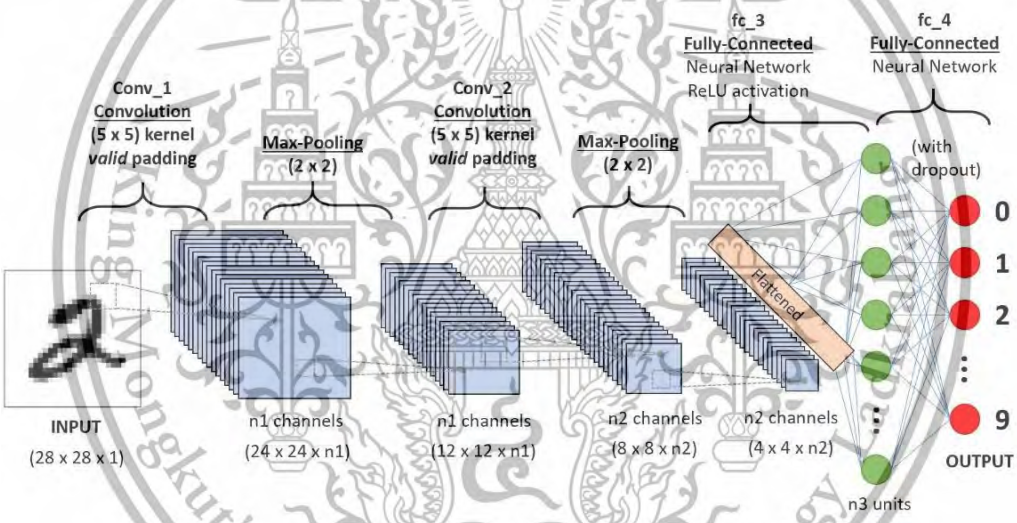

Figure 2.2 CNNs Architecture[4]

2.3.2 Transfer Learning (Base Model as Feature Extraction)

Transfer learning [5] is a DL technique where a model developed for one task is reused as the starting point for a model on a second task. This process is highly effective in cases where the data available for the new task is limited. Transfer learning can significantly reduce training time and computational resources required for training a new model from scratch.

For example, a model pre-trained on ImageNet, a large-scale dataset for object recognition, already knows how to extract features from images. The lower layers of

the model have learned to recognize textures, shapes, and patterns, while the higher layers have learned more complex representations. By removing the last layer (or layers) and adding new layers tailored for brain tumor classification, the model can be fine-tuned on a smaller dataset of brain tumor images.

This approach of using a base model as a feature extractor is particularly effective in the context of brain tumor classification because the pre-trained model already has the ability to extract relevant features from images. The new layers added to the model can then learn the specific features associated with brain tumors. By using transfer learning, models can achieve high accuracy in brain tumor classification, even when the available data is limited.Transfer learning also provides the advantage of stability. Since the pre-trained model has already learned robust features from a large dataset, the model is less likely to overfit the new, smaller dataset. This stability leads to better generalization and improved performance on unseen data. The flow of transfer learning process is shown in Figure 2.3

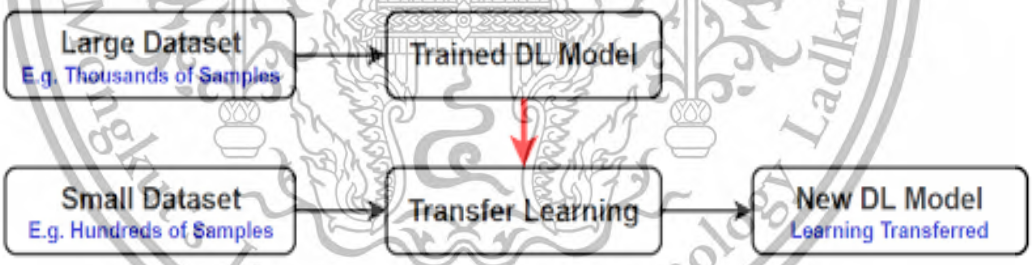

Figure 2.3 Tranfer Learning diagram[6]

2.3.3 ResNet50V2

ResNet50V2 is a network architecture that was presented in the research paper titled "Identity Mappings, in Deep Residual Networks" authored by Kaiming He, Xiangyu Zhang, Shaoqing Ren and Jian Sun [7]. The main concept behind the ResNet (Residual Network) architecture lies in incorporating "shortcut" or "skip" connections. These connections allow the network to bypass layers, which helps tackle the issue of vanishing gradients often encountered in networks. A significant enhancement

introduced in ResNets V2 variant is pre activation. Under this approach batch normalization and Rectified Linear Unit (ReLU) activation are applied before convolution takes place. This modification has been proven to result in accuracy. The term "50", in ResNet50V2 denotes the depth of the network indicating 50 layers.

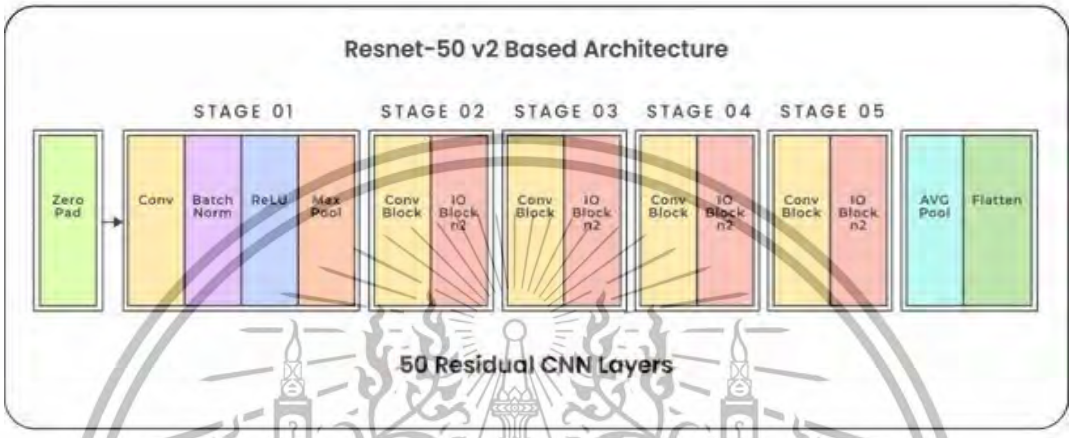

Figure 2.4 ResNet50V2's architecture[8]

2.3.4 InceptionResNetV2

The InceptionResNetV2 structure combines two neural network designs; the Inception network and the Residual Network. It was introduced in a paper called "Inception v4, Inception ResNet and the Impact of Residual Connections, on Learning" by Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke and Alex Alemi [9]. The original Inception network, known as GoogLeNet introduced the concept of Inception modules. These modules involve operations with receptive fields (like 1x1, 3x3 and 5x5 convolutions) that are then concatenated together. This allows the model to independently determine the filter sizes. The InceptionResNetV2 combines these inception modules with the connections from ResNet. One important improvement, in this combined architecture is scaling down the residuals before adding them to the signal, which enhances the training process.

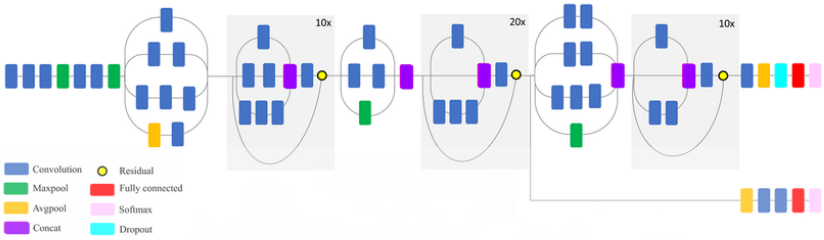

Figure 2.5 InceptionResNetV2 architecture [10]

2.3.5 DenseNet121

The architecture of DenseNet121 is quite advanced when it comes to using parameters and improving the spread of features. To start the model takes an input image. Passes it through a layer, with a 7x7 kernel and a stride of 2. Then a max pooling layer is used to reduce the dimensions. The key aspect of this design is the blocks where each layer is connected to every layer. The feature maps from all layers are. Passed on to subsequent layers. This combination process ensures reuse of features, throughout the network, which in turn improves flow during training.

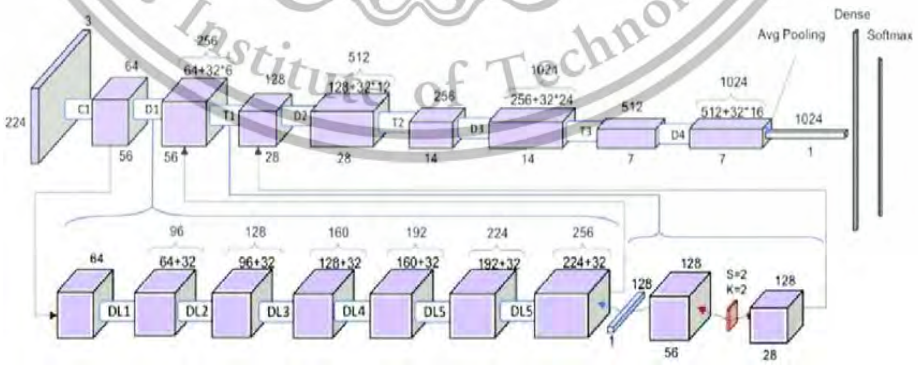

Figure 2.6 DenseNet121's architecture [11]

2.4 Preprocessing of MRI image

Accurate classification of brain tumors heavily relies on the preprocessing of MR images. Since the raw images can have variations, in intensities and qualities it becomes crucial to standardize them for analysis and modeling purposes. Below show section of the steps employed in this research to prepare the images for classification.

2.4.1 Normalization

To ensure consistency and comparability one of the steps is to standardize the images. Since raw MRI scans may have varying intensity ranges that do not fall within [0, 255] the first task is to adjust them. Using the normalize technique we modify the images so that their intensities now fall within the range of [0, 1]. This step plays a role, in maintaining uniformity and ensures that each pixel value, in every image has an impact when inputted into the model. The normalization is calculated using equation(2.1)

$$X_{norm} = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (2.1)$$

2.4.2 Median Filtering

Once the images are normalized we employ the OpenCV library to apply filtering. This step plays a role, in eliminating noise that often exists in MR scans. By utilizing a 3x3 kernel size the median filter substitutes each value with the value of its surrounding pixels. This process effectively smooths out the image while preserving its edges intact. The objective is to enable the model to learn from image features than being influenced by random noise.

2.4.4 Conversion to RGB

While MR images are inherently grayscale, many DL architectures, especially those pre-trained on datasets like ImageNet, are designed to accept 3-channel red,green,and blue (RGB) images. To make the images compatible with such architectures, each normalized, filtered grayscale image is replicated across three channels to form a pseudo-RGB image. This step does not introduce any new information to the image, but simply adapts it for compatibility with a broader range of neural network architectures.

2.5 Data Augmentation in Medical Imaging

Medical imaging datasets inherently have limitations. Unlike domains where data can be freely collected or scraped medical images require expert annotations adhere to ethical guidelines and are often difficult to obtain. Moreover certain pathological findings may be rare leading to imbalanced datasets with conditions. In scenarios using the dataset alone can result in overfitting models that perform exceptionally well on training data but struggle with generalization on unseen data—a significant drawback in a medical context where false predictions can have serious consequences. Data augmentation serves as a solution, by increasing the size and diversity of the training dataset providing the model with a perspective and enabling better generalization.Data augmentation, in imaging involves techniques that enhance the performance of DL models.

These techniques include;

- Rotation: In medical imaging, ensuring a model's invariance to image orientation is crucial. This is particularly true as scans may be captured from different angles or orientations due to patient positioning or other procedural variations. Thus, rotating images during augmentation introduces variations mimicking these different capture angles. By training on these rotated images,

the model becomes robust, understanding the underlying features irrespective of their orientation in the image space.

- Zooming: Features of interest in medical images might present themselves at various scales. For instance, a lesion or tumor might be relatively small in one image and substantially larger in another. By employing zooming techniques — either zooming in to magnify specific features or zooming out to reduce them — the model is conditioned to recognize and understand these features at various scales. This ensures that even if a particular feature appears at a different size in a new image, the model's performance remains consistent.
- Shearing: The shape and alignment of anatomical structures can differ based on the angle of imaging or physiological differences among patients. Shearing, as an augmentation technique, displaces certain points of an image in a fixed direction, effectively altering the shape of structures within. When the model is exposed to these modified shapes during training, it learns to recognize similar structures even if they appear slightly sheared or misaligned in real-world scenarios.
- Flipping: The human anatomy is bilaterally symmetrical, and certain conditions can manifest similarly on either side of the body. By introducing both horizontal and vertical flipping as augmentation techniques, we present the model with varying perspectives of viewing. This ensures the model remains adept at identifying features even if they appear mirrored or flipped, simulating real-world scenarios where conditions manifest on either side or when images are captured from a flipped perspective.

- Brightness and Contrast Adjustments: Variability in imaging due to equipment differences, settings, or ambient conditions can introduce variations in brightness and contrast across medical images. It's not uncommon for two images, captured using different machines or settings, to exhibit stark contrast or brightness differences.

The use of data augmentation has been proven effective in improving DL model performance for imaging tasks. Training on a set of augmented images enhances model robustness. Reduces overfitting risks. Augmented data provides context, for better feature extraction. Ultimately leads to improved accuracy. Training models using augmented data has shown performance, on data, which is particularly important in medical imaging. In this field models may be utilized in settings with different imaging devices and patient populations. To address the challenge of images for conditions data augmentation can help ensure a balanced dataset preventing the model from being biased towards the majority class.

Data augmentation acts as a bridge, between the scarcity of images and the data requirements of learning models. By expanding the datasets size and diversity it not ensures theoretical accuracy but also enhances clinical relevance and prepares models to tackle real world challenges.

Figure 2.7 Data augmentation on MRI image

2.6 Model SOUPs Including Uniform Soup and Greedy Soup

Model SOUPs, which include Uniform Soup and Greedy Soup have emerged as approaches, in DL. They aim to enhance model performance by leveraging insights from models. Model SOUPs or "Model Selection with Performance " involve selecting models to make predictions [12].

The core idea behind Model SOUPs is to create a "soup" of model weights derived from trained models and then averaging them. By doing this approach combines the strengths of models to generate a more robust and accurate final model. Unlike techniques Model SOUPs do not incur additional inference or memory costs since the averaged weights are used for inference.

There are two techniques for creating Model SOUPs; Uniform Soup and Greedy Soup.

- Uniform Soup; In this technique all the trained model weights are equally averaged regardless of each models performance, on the held out validation set. This method assumes that every model contributes equally to the performance. The advantage of Uniform Soup lies in its simplicity and ease of implementation. However it may not always yield results since it does not consider models performances.

- Greedy Soup; In the Greedy Soup method we start by arranging the trained models, in descending order based on their performance on the validation set. Then we gradually add weights to the performing model only if it enhances the performance of the soup. This technique aims to maximize the influence of performing models while minimizing the impact of performers. Compared to Uniform Soup Greedy Soup tends to yield outcomes as it takes into account individual model performance.

When it comes to brain tumor classification Model SOUPs can be highly effective. Since tumor appearances in images can vary significantly among patients combining insights from models through Model SOUPs helps capture a wider range of

tumor appearances and leads to improved classification performance. The Model SOUPs approach proves beneficial when dealing with training data. By utilizing models this "soup" effectively captures a spectrum of features and representations reducing overfitting risks and enhancing generalization capabilities towards new and unseen data.

To sum up Model SOUPs provide an efficient approach for enhancing model performance, in brain tumor classification. By utilizing the knowledge derived from models and finding the average of their influences Model SOUPs can generate dependable and precise forecasts thereby improving the trustworthiness of brain tumor categorization. Essemble model selected by using equation in Figure 2.9

	Method	Cost
Best on val. set	$f(x, \arg\max_i \mathsf{ValAcc}(\theta_i))$	$\mathcal{O}(1)$
Ensemble	$\frac{1}{k}\sum_{i=1}^{k} f(x, \theta_i)$	$\mathcal{O}(k)$
Uniform soup	$f\left(x, \frac{1}{k}\sum_{i=1}^{k} \theta_i\right)$	$\mathcal{O}(1)$

Figure 2.8 Equation of Model soup [12]

2.7 Five folds cross validation

To implement a five-fold cross validation for classifying brain tumors, an approach that ensures the evaluation of the model is robust needs to be followed. First and foremost it's crucial to clean and preprocess the dataset of brain MR images. This involves tasks like normalization and resizing to prepare the data optimally for training. Once the dataset is prepared it is divided into five subsets or "folds". It is important that each fold represents the dataset adequately and includes a representation of each class.

The essence of validation lies in iterative training and validation. In each iteration the classification model is trained using data from four out of five folds while

rotating which fold serves as the validation set. After training on these four folds the model is then validated on the remaining fold to ensure testing on unseen data. The performance metrics such, as accuracy, precision, recall or F1 score are recorded for analysis.

Once all five folds have been used for validation their results are compiled together. Calculating performance across all validations provides an understanding of how well the model performs overall.

Moreover analyzing the variance or standard deviation of the metrics, across the validations offers insights into how consistent the model performs. A model with variance is preferred as it suggests reliability when dealing with unseen data.

Once we've employed 5 cross validation to evaluate the models performance and optimize hyperparameters we proceed with a final training phase. This phase utilizes the dataset to ensure that the model is perfected before deployment or making predictions.

The true value of employing 5 cross validation in brain tumor classification lies in its robustness. Not does it provide a performance estimate through multiple validation phases but it also helps address overfitting concerns. This approach guarantees that each data point is utilized for both training and validation purposes at stages maximizing data efficiency. The iterative nature of this method also facilitates tuning hyperparameters establishing a framework, for evaluating and refining models.

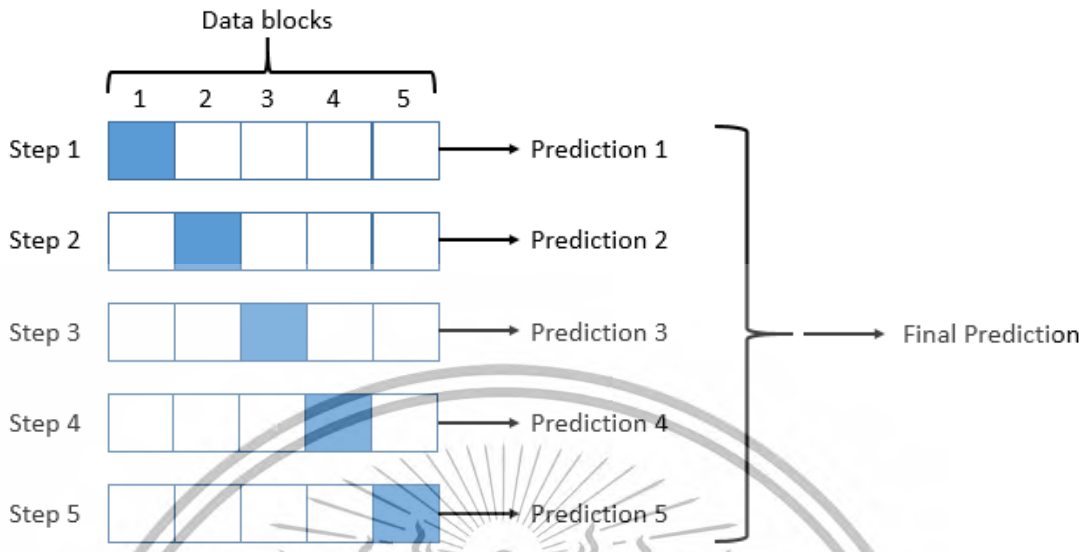

Figure 2.9 K-Fold Cross Validation process [13]

2.8 Evaluating Model Performance

Evaluating the performance of a model is crucial when it comes to classifying brain tumors. Accurately identifying tumors plays a role, in diagnosis and treatment directly impacting the well being of patients. Lets explore some metrics and their significance in evaluating model performance for brain tumor classification;

Accuracy; This metric is pretty straightforward as it represents the proportion of classified instances out of the instances. However in datasets where there is an imbalance between classes accuracy can be misleading. A model that only predicts the majority class will still have accuracy but won't be helpful.

Confusion Matrix; The confusion matrix is a table used to describe how well a classification model performs on a set of data with known values. It provides a breakdown of incorrect classifications making it easy to identify where exactly the model excels or struggles. The confusion matrix consists of four values; True Positives (TP) False Positives (FP) True Negatives (TN) and False Negatives (FN) which can be calculated by using equation shown in Figure 2.10.

	Reference standard positive	Reference standard negative
Index test positive	*True positive (tp)*	*False positive (fp)*
Index test negative	*False negative (fn)*	*True negative (tn)*

Accuracy measures:

- *Sensitivity: tp/(tp+fn)*
- *Specificity: tn/(tn+fp)*
- *Positive predictive value: tp/(tp+fp)*
- *Negative predictive value: tn/(fn+tn)*
- *Diagnostic odds ratio: (tp x tn)/(fn x fp)*

Figure 2.10 Confusion matrix and accuracy measures [14]

Accuracy is the measure of all correct predictions made by the model over all kinds of predictions. It gives us a straightforward indication of how often the model is right.

Precision and Recall; Precision measures the proportion of predictions, among all positive predictions essentially indicating how many instances that were labeled as positive are actually positive.

Recall is, about how well the model can find all the instances among all the actual positive predictions. It tells us how good the model is at identifying tumors when they're present. On the hand precision is about how likely it's for the model to be correct when it predicts a tumor. So high precision means that when the model predicts a tumor it's more likely to be correct. High recall means that the model can identify most of the tumors that're actually there.

The F1 score combines both precision and recall into a metric. Gives us a balanced measure of performance. It comes in handy when we have an imbalance in class distribution. We need one metric that considers both precision and recall together. A

higher F1 score nearer to 1 indicates higher performance while a lower F1 score closer to 0 indicates poorer performance. The accuracy, precision,
recall, and F1-score are evaluated in Equations (1-4) [15]

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN} \quad (1)$$

$$Precision = \frac{(TP)}{(TP + FP)} \quad (2)$$

$$Recall = \frac{(TP)}{(TP + FN)} \quad (3)$$

$$F1 - score = \frac{2\,x\,(precision + Recall)}{(precision + Recall)} \quad (4)$$

2.9 Streamlit Bringing Medical Imaging to the Web

In a time where ease of access and real time insights are highly valued it becomes crucial to have a user platform, for deploying DL models, especially those related to medical imaging. That's where Streamlit comes in. Streamlit is an open source app framework that has gained popularity among data scientists and developers. It is specifically designed for DL and data centric applications providing a way to transform MRI based brain tumor classifiers or any other model into interactive web applications.

What makes Streamlit stand out is its simplicity. With a few lines of Python code data script can turn into web application. For professionals this means they can easily upload MRI images and receive predictions directly in their web browser without needing to understand the complexities of the underlying model.

Streamlit offers a range of widgets including sliders, buttons and file uploaders. These widgets are particularly useful, in the field of imaging.For example radiologists have the ability to adjust settings or thresholds upload MRI scans. Even visualize the layers and activations of a neural network using simple interactions, on the application.

The flexibility and scalability of Streamlit are its strengths. By being compatible with Python libraries and tools developers can incorporate visualizations utilize advanced algorithms and seamlessly integrate with databases or cloud solutions. This adaptability makes it a versatile platform for medical imaging solutions.

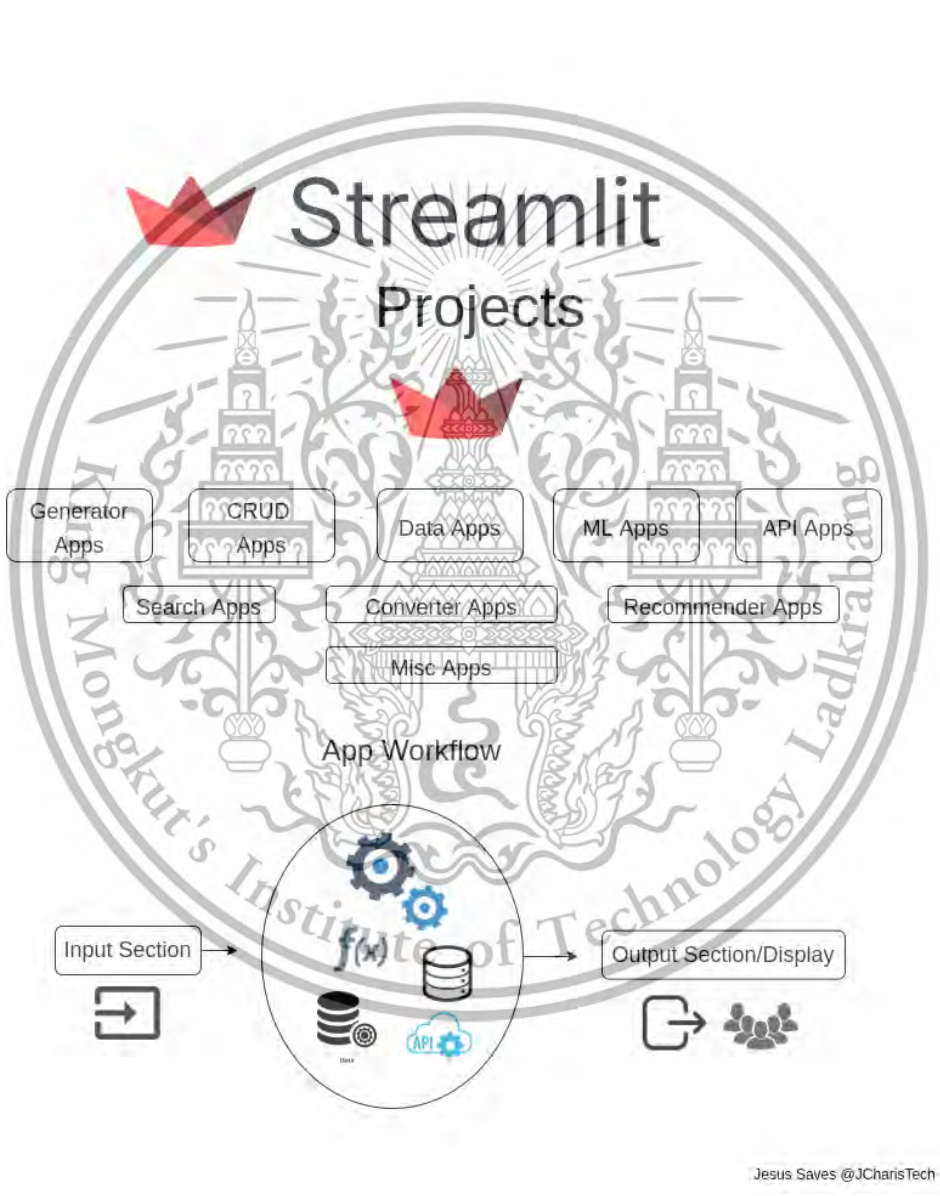

Figure 2.11 Application of Streamlit [16]

2.10 Related researched paper

In recent literature, a notable study focused on the classification of brain tumor MR images, specifically targeting three types: meningioma, glioma, and pituitary. Additionally, the research went a step further to classify gliomas into distinct grades, namely Grade II, Grade III, and Grade IV. The study utilized two datasets, with the first encompassing 3064 T1-weighted contrast-enhanced images from 233 patients, and the second containing 516 images from 73 patients.

The proposed CAD (Computer-Aided Diagnosis) system in this study is underpinned by a custom deep neural network comprising 16 layers. This architecture includes three convolution layers, each followed by a ReLU activation function, normalization, and max pooling. To mitigate overfitting—often a challenge in DL models—two dropout layers were incorporated. The network culminates in a fully connected layer, a softmax layer for output prediction, and a classification layer that produces the predicted class. Recognizing the limitation of the dataset's size, data augmentation techniques were employed to enrich the training samples, ultimately enhancing model performance. Remarkably, the system achieved an impressive accuracy of 96.13% and 98.7% for the two datasets, respectively [17].

One of the pivotal challenges in modern healthcare, especially within IoT-healthcare systems, is the classification of brain tumors. The precise diagnosis of brain cancer can significantly benefit from the integration of AI into the diagnostic process. However, a prevailing concern has been the accuracy of such AI-based diagnostic systems. This sentiment has sometimes made the medical community hesitant to wholly embrace these technologies.

A recent study aimed to bridge this gap by proposing a deep learning-based approach that utilizes an enhanced Convolutional Neural Network (CNN) model for classifying brain tumors. The focal point of this study is the classification of three primary tumor types: Meningioma, Glioma, and Pituitary, using brain MR images. What distinguishes their approach is the strategic integration of transfer learning and data augmentation techniques to bolster the predictive capability of the CNN model. The integrated diagnostic framework, termed ResNet-CNN, exhibited outstanding

performance, boasting an accuracy rate of 99.90%. This surpasses many baseline methods and signifies a marked improvement in the field. Several factors contributed to this model's high predictive outcomes: meticulous data preprocessing, astute parameter adjustments (such as layer numbers, optimizer choices, and activation functions), and the judicious employment of transfer learning and data augmentation. Given its stellar performance, the research proposes that the ResNet-CNN model holds immense potential for brain tumor classification in IoT-Healthcare systems [18].

2.11 Chapter Summary

In this chapter we have delved into an exploration of the diverse world of brain tumor classification. We began our journey by discussing the anatomy of brain tumors. Identified four main types; glioma tumors, meningioma tumors, pituitary tumors and healthy brain tissue, as a benchmark. We explored their characteristics and origins highlighting the importance of distinguishing these categories for patient prognosis and treatment strategies.

Moving from biology to technology we immersed ourselves in the algorithms that are shaping the future of imaging. Our discussion emphasized CNNs as players in image tasks like MRI based tumor classification. Furthermore we highlighted the potential of Transfer Learning to revolutionize our classification efforts by leveraging trained models such as ResNet50V2 and InceptionResNetV2. These models have gained knowledge from datasets ImageNet providing us with a wealth of learned features.However it is worth noting that challenges exist within the field of imaging.

Like any endeavor it is crucial to measure our success. In our chapter we dedicated a portion to evaluation metrics highlighting the need, for not accuracy but a comprehensive set of measures including precision, recall, F1 score and more. Additionally we introduced Streamlit as a groundbreaking technology that brings together imaging and web based interactivity.

As we move forward into the following chapters our goal remains clear. To utilize these state of the art techniques and create a brain tumor classification system in terms of accuracy and practicality.

CHAPTER 3
METHODOLOGY

3.1 Introduction

In Chapter 2 we explored the foundations of diagnosing brain tumors and how AI can enhance methods. This chapter moves from theory to application by explaining the approach we took in developing an AI based system for classifying brain tumors. We start by examining our design process discuss the obstacles we faced along the way and conclude with presenting our proposed solution.

3.2 Design Methodology

Grounded in the theoretical insights from Chapter 2, our approach to designing the brain tumor classification system is systematic. Let's examine how the design decisions were influenced by the theories and techniques discussed in Chapter 2.

3.2.1 Data Collection & Data prepataion

The study collected data, from Kaggle, a source of brain tumor images. The dataset comprises 3064 T1 weighted contrast enhanced images from 233 patients categorized into three types of brain tumors; meningiomas (708 slices) gliomas (1426 slices) and pituitary tumors (930 slices). This comprehensive classification showcases the datasets nature. Researchers have widely trusted this dataset for its relevance and credibility in research projects. For instance Cheng et al [19]. in their publication titled "Enhanced Performance of Brain Tumor Classification via Tumor Region Augmentation and Partition" (2015) referred to this dataset to support their findings. Moreover Cheng et al. Also utilized the dataset in another work called "Retrieval of Brain Tumors by Adaptive Spatial Pooling and Fisher Vector Representation" (2016) [20]. The fact that these studies utilized this dataset highlights its quality and significance in the field of brain tumor research.

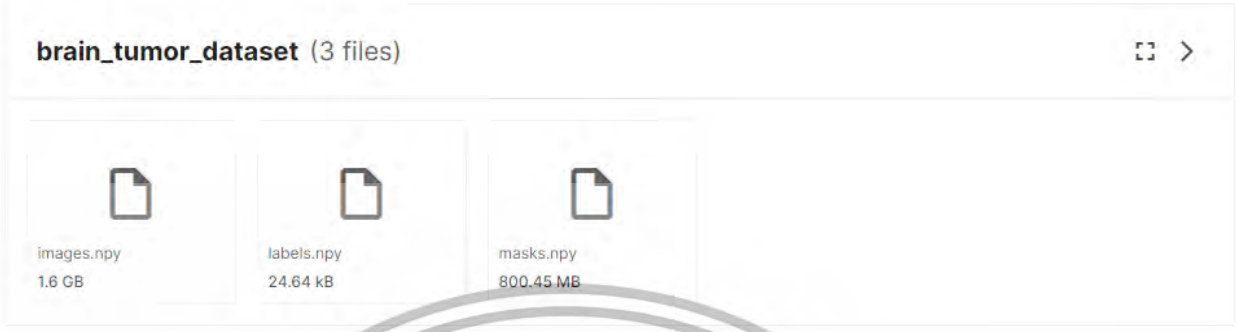

Figure 3.1 Brain_tumor_dataset in Kaggle

After obtaining the data our immediate focus was on preprocessing it to ensure readiness, for stages. We performed preprocessing on each image within the dataset by converting the image data type to float32. This conversion ensures consistency. Enables calculations throughout the stages.

First we normalized the image by using its maximum and minimum values. This step is crucial as it ensures that the pixel values are, within a range, which improves the stability and convergence rate of the model during training. Next we applied a filter to reduce any noise in each image. We opted for filters as they effectively eliminate salt and pepper noise resulting in higher quality images. Finally even though MRI images are typically grayscale we converted the processed image to RGB format to ensure compatibility with DL architectures that expect input images with three channels. Once the preprocessing was complete we saved the conditioned images as Joint Photographic Expert Group (JPG) files to streamline processing stages and minimize any bottlenecks.

Figure 3.2 Processed_Image

After completing the step of preprcoessing we divided our dataset into subsets for training, testing and validation purposes. This division of data is crucial for our DL models as it allows us to train, validate and test them on datasets. By doing we can reduce overfitting. Improve our models ability to generalize well. Specifically we allocated 70% of the data for training purposes while evenly distributing the remaining 30% between validation and testing sets.

One important aspect of our data preparation involved creating two training sets. The first set maintained the class distribution. Allowed us to perform, on the fly data augmentation during training. This dynamic augmentation strategy introduces real time variability into our training process. Enhances our models adaptability when faced with data.

On the hand we took care in crafting the second training set by adding more examples to the minority classes. Our objective was to tackle any imbalances, between classes and ensure that the model doesn't favor the majority class during training.

3.2.3 Model Design and Choice

Our effort to create a solution, for classifying brain tumors involved using three regarded DL architectures; DenseNet121, ResNet50V2 and InceptionResNetV2. These architectures have gained recognition in the field of computer vision have shown performance in various tasks. Below we provide details about the composition and setup of each model;

For DenseNet121's architechture showing Figure 3.4. We defined the input shape as (224, 224 3). Excluded the layer of the model. This allowed us to add custom layers tailored for our task. We initialized the weights using the known ImageNet dataset to benefit from trained knowledge. Our composition for DenseNet121 included feature extraction through an pooling layer that gathered feature maps into a coherent structure. To prevent overfitting and ensure regularization we added a dropout layer with a rate of 0.5. Lastly we included a dense layer with 3 output units representing our classes and used softmax activation function to determine class probabilities.

As for ResNet50V2' which is recognized for its connections addressing the vanishing gradient problem we employed it as another architecture in our project. With the input shape, as before and weights initialized from ImageNet dataset our ResNet50V2 model consisted of feature extraction using ResNet50V2 followed by average pooling layer. We used a layer with 1024 units and a ReLU activation function followed by a dropout layer, with a rate of 0.3. Then we added another layer with 512 units activated by ReLU and another dropout layer with a rate of 0.2. Finally we included an output layer with 3 units and a softmax activation function to determine our class probabilities. The overall architechture of ResNet50V2 is shown in Figure 3.3

To create our model based on the InceptionResNetV2 architecture Figure 3.5 we combined the strengths of the inception architecture with the connections of ResNet. Our model included feature extraction using InceptionResNetV2 followed by pooling

and a dropout layer at a rate of 0.55. We then added a layer with 60 units activated by Exponential Linear Unit (ELU). Initialized using the GlorotNormal method along with another dropout layer at a rate of 0.3. The final dense layer consisted of 3 units activated by softmax.

For all three architectures mentioned we started with weights derived from the ImageNet dataset—a large scale dataset used for object detection purposes. By utilizing these pretrained weights our models had an advantage in recognizing patterns and textures in images during training resulting in more effective classification for our specific problem.

In conclusion we chose these architectures based on their proven effectiveness, in image classification tasks.The models were subsequently. Customized to meet the distinct needs of brain tumor classification.

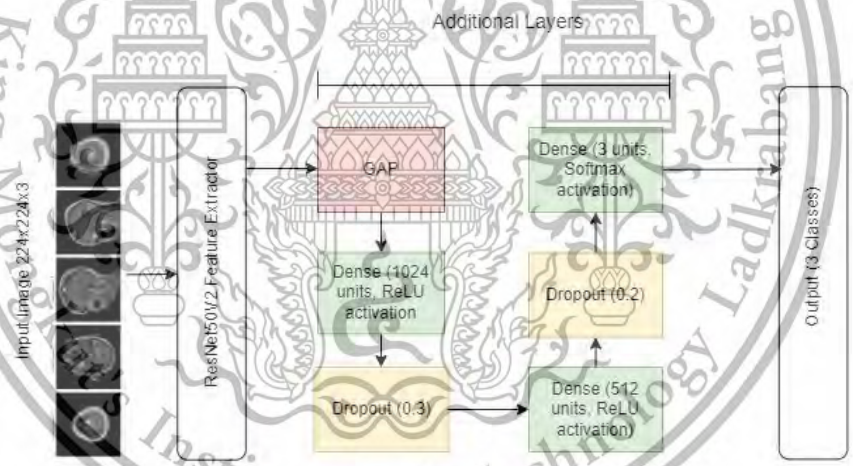

Figure 3.3 Fine-Tuned ResNet50V2

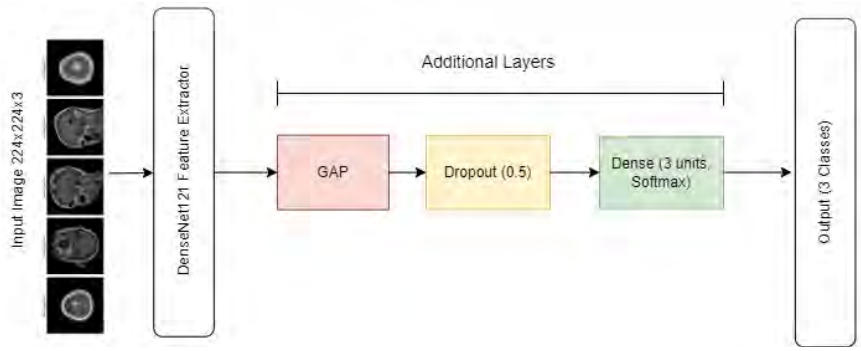

Figure 3.4 Fine-tune DenseNet121

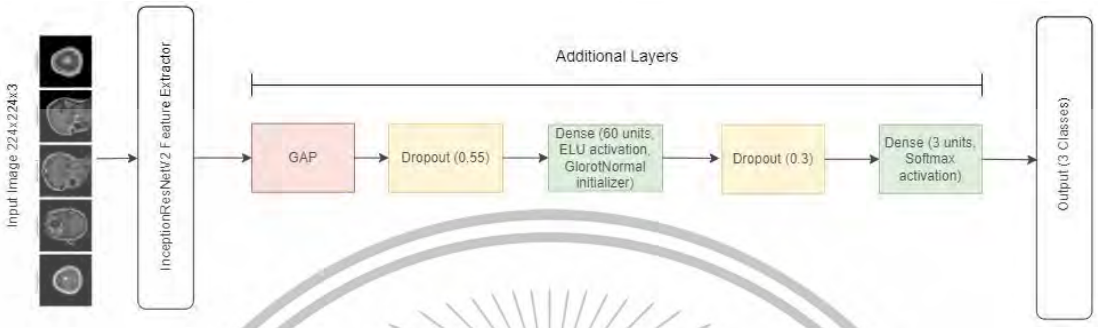

Figure 3.5 Fine-tune InceptionResNetV2

3.2.4 Five-Fold cross validation

To incorporate five fold cross validation into our project we initially divide the data into two sets; a combined training/validation set and a separate test set. The test set is kept aside. It will only be used for the final evaluation. For the validation process as shown in Figure 3.6 we further split the combined training and validation set into five "folds". During each iteration four of these folds (indicated in green) are utilized to train the model while the remaining fifth fold (shown in blue) is used for validation purposes to check the models performance.

Following each iterations training phase we can evaluate the resulting models performance using the reserved test set. This provides us with an accuracy score. Loss metric for every iteration. Once all five iterations are completed we will eventually have 5 models from different iteration. Finally we calculate the mean test accuracy and standard deviation across all iterations.

With the insights gained from the mean accuracy and standard deviation, we can make more informed decisions about next experiments which is the model soup.

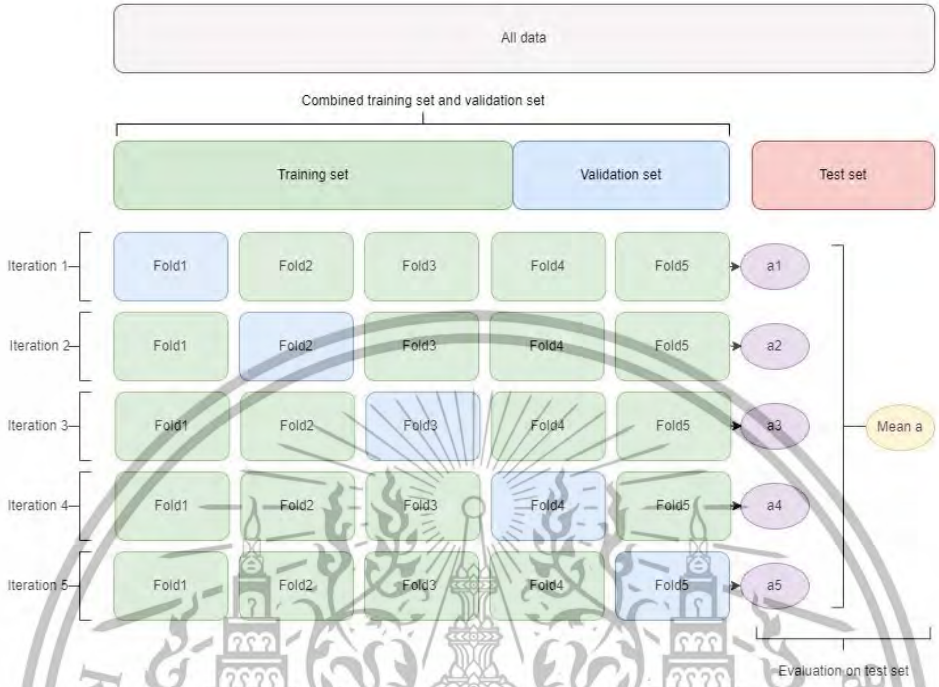

Figure 3.6 Process of Five-Fold cross validation

3.2.3 Training Process

The training process of individual model show in Figure 3.7 embarked on a comprehensive exploration of hyperparameters, optimizing them to extract the most performance out of our chosen architectures. The fine-tuning of these hyperparameters is pivotal as they wield substantial influence on the learning dynamics and, subsequently, on the model's performance.

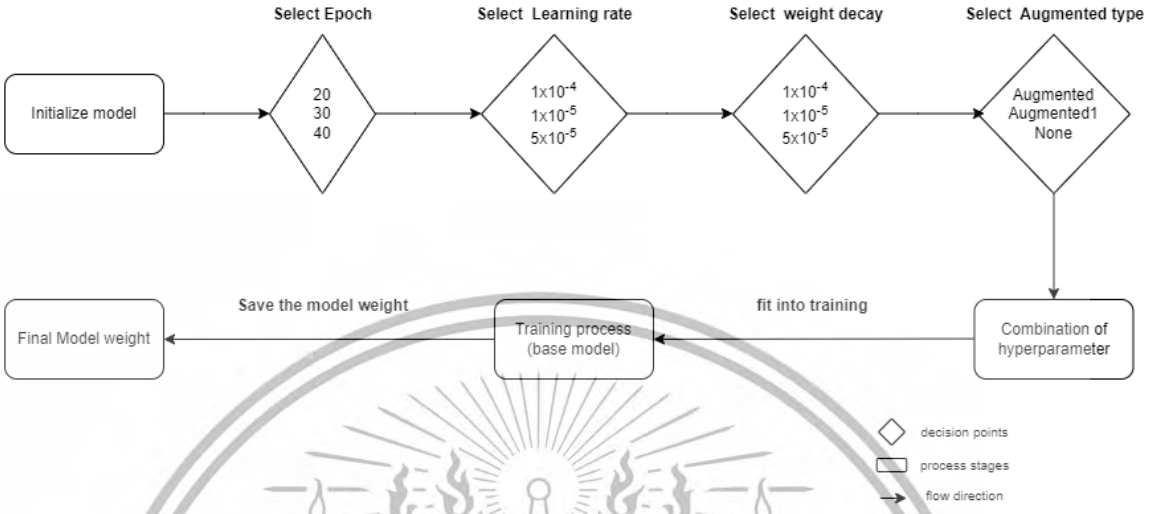

Figure 3.7 Training process

Hyperparameters:

- Epochs: The models were trained for varying epochs - 20, 30, and 40. The term 'epoch' refers to one complete forward and backward pass of all the training examples. The more epochs we run, the more the model improves, up to a certain point. After that, the model can start overfitting.
- Learning Rates: Different learning rates were trialed $1 \times 10^{-4}, 1 \times 10^{-5}, 5 \times 10^{-5}$ The learning rate dictates the size of the steps the model takes to adjust its weights while minimizing the loss. Too large a rate might overshoot the optimal point, while too small a rate could lead to exceedingly slow convergence.
- Weight Decay: Weight decay rates of $1 \times 10^{-4}, 1 \times 10^{-5}, 5 \times 10^{-5}$ were explored. Weight decay acts as a regularization method, preventing the weights from growing too large and, thus, combating overfitting.
- Three distinct training paradigms were adopted. The first two, termed augment and augment1, involve 'on-the-fly' data augmentations, a strategy to artificially boost the diversity of data available for training without collecting new data. These augmentations introduce minor alterations to the original dataset,

enhancing the model's generalization capabilities. The third paradigm, denoted as None, utilized the pre-augmented dataset for training. This means the data was augmented beforehand and stored, ensuring a consistent set of augmented data throughout the training.

Furthermore, we incorporated an Early Stopping mechanism. This essential tool monitors a specified metric (like validation loss) during the training phase. If the metric ceases to improve after a set number of epochs, training is halted, thus preventing potential overfitting and saving computational resources.

Training Paradigm

We implemented a training process by training each architecture (base model) using combinations of the mentioned hyperparameters. This resulted in a total of 20 sets of model weights, for each base model. These weights represent aspects of the training process potentially optimized for aspects of the problem.

After this training we used two strategies called Greedy Soups and Uniform Soups. These approaches combine the strengths of models to create a robust and stable solution. We applied the weights obtained from these soups to our models, which were then tested using the same base architecture. We compared their performance with each of individual models to evaluate how effective the ensemble approach is compared to individual models.

Ultimately our rigorous training methodology and ensemble strategies aimed not to tune the models, for the specific task but also to ensure their resilience when faced with new and unfamiliar data challenges.

3.2.3.2 Greedy soup

The "Greedy Soup" method starts by initializing the model with the weights of the best performing individual model. This sets a performance, for iterations. As we move forward we sequentially load the weights of each model. Combine them with the

current ensemble weights. This creates a version of the ensemble often called a "soup". We then evaluate the performance of this ensemble to determine its effectiveness.

At each iteration an important decision is made; if the created "soup" shows performance compared to the previous version we adopt this new combination of weights for the ensemble. On the hand if there is no improvement or if it worsens we go back to using the weights from the best performing version. This means that we discard any contribution from models that didn't improve our results.

This iterative process continues until all models weights have been assessed. By the end of this process using the "Greedy Soup" method ensures that our saved ensemble model is superior, to all combinations considered. It's worth noting that if no combination of weights outperforms our model during evaluation we default to using just that best individual model.

Figure 3.8 Process of greedy soup

3.2.3.1 Uniform soup

The "Uniform Soup" technique is a effective method of combining multiple models to leverage their collective intelligence. Unlike the "Greedy Soup" approach, which focuses on selecting weight combinations, for performance the "Uniform Soup" method operates on the principle of equality. It starts by taking all model weights and uniformly combining them ensuring that each model contributes equally to the

ensemble. There is no bias towards any models performance; every model has a say in the final ensemble.

After combining the weights we evaluate the resulting ensemble or "soup" to measure its performance. The main advantage of this approach is its simplicity and the belief that a balanced combination of models can capture a range of patterns and insights from the data. While it may not always outperform ensembling techniques, like the "Greedy Soup," the "Uniform Soup" provides a democratic and less computationally intensive alternative. It recognizes that each models perspective holds value and assumes that when aggregated together these perspectives can generate reliable predictions.

Figure 3.9 Process of Uniform soup

3.3 Interesting Problems

The field of DL learning has made significant advancements and brought about transformative changes, in various industries. Healthcare in particular has greatly benefited from these advancements. One area where neural networks have shown promise is in analyzing medical images like MRI. However like any emerging technology there are challenges to overcome. Some of these challenges are unique to the field of imaging while others are common across applications of deep learning.

In this section we will explore some of the challenges that arise during projects involving medical image analysis. By understanding these complexities we not gain insight into the intricacies involved in analyzing images but also emphasize the importance of meticulous attention, to detail when effectively utilizing DL in a clinical setting.

3.3.1 Small Size of Dataset

In the field of imaging the issue of having datasets often arises especially, due to the sensitive nature of medical data and the stringent procedures involved in its collection. The medical industry adheres to legal standards to safeguard patient privacy. Additionally since many medical conditions are rare and not frequently documented small datasets are quite common.

The consequences of working with a dataset are numerous. Firstly there is a risk of overfitting. Overfitting happens when a model becomes too specialized in capturing every detail of the training data, which ultimately hampers its ability to effectively generalize to unseen data. Consequently this leads to performance in real world scenarios. Furthermore small datasets pose challenges, for conducting validation and testing processes that accurately assess a models ability to generalize effectively.

3.3.2 Diversity View of MRI Image

MRI has completely transformed the field of imaging by providing a invasive approach and delivering incredibly detailed information. However the multi dimensional and high resolution nature of MRI images also brings about a range of variations. These variations can be attributed to factors such, as the brand, model, calibration and even software versions of the MRI machine being used. Additionally differences in positioning, physiological characteristics and scan protocols contribute to this diversity.

Managing diversity is crucial because for a model to be useful in a setting it needs to perform consistently under various conditions and scenarios. Even the slightest variation, like taking an MRI scan from a angle or position can lead to significantly different outcomes. This poses a challenge when it comes to maintaining performance, for the model.

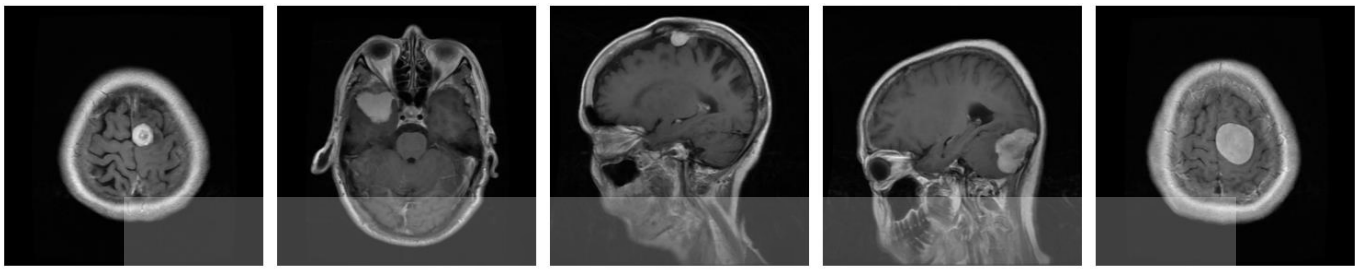

Figure 3.10 Diverse view of MRI Image

3.3.3 Imbalance Classes

Class imbalance poses a challenge, in DL applications and it can have particularly detrimental effects in the field of medical imaging. The real world occurrence of diseases or conditions can naturally result in their underrepresentation within datasets. For instance while one type of tumor might be commonly diagnosed others may be quite rare.

When models are trained on imbalanced datasets like these they tend to lean towards predicting the majority class due to its prevalence, in the training data. This leads to a model that consistently misclassifies the minority class. In a context such misclassifications can have consequences.

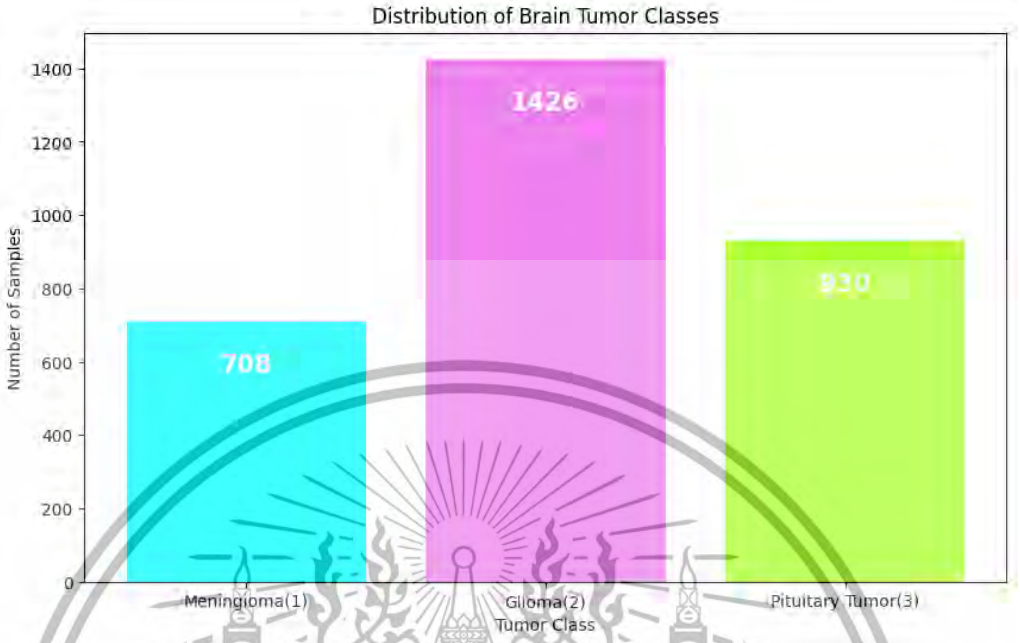

Figure 3.11 Distribution of Brain Tumor classes

3.3.4 Computational cost

The computational demands of learning models can be quite substantial. Models, like DenseNet, ResNet or InceptionResNetV2 are well known for their accuracy but infamous for requiring resources. Training these models not requires computers but also extensive memory often necessitating specialized hardware such as Graphics processing unit (GPU) or Tensor Processing Unit (TPU).

3.5 Proposed Solution

In our pursuit of understanding learning using MRI images we encountered obstacles as mentioned earlier. However challenges often serve as catalysts, for innovation. In this section we will provide an overview of the solutions we developed to overcome these complexities.

3.5.1 Small Size of Dataset

To counteract the limited dataset size, we employed augmentation. By introducing random transformations such as rotations, translations, and zooming on our existing images, we expanded and diversified our data pool. This augmented data not

only provided the models with more learning material but also improved their generalizability.

3.5.2 Diversity View of MRI Image

Diversity in MRI image perspectives, while essential, introduced an element of variability. Our strategy here was to balance the dataset. By ensuring that each MRI view was equally represented, we ensured a uniform and comprehensive training and promoting consistency in predictions.

3.5.3 Addressing Imbalance Classes

To address the challenge of data imbalance, we formulated two distinct training sets. The first set is pre-augmented, ensuring a balanced representation across all classes. The second set retains its original, unbalanced composition. By diversifying our training sets in this manner, we aim to assess the performance variations due to data augmentation and class balance more effectively.

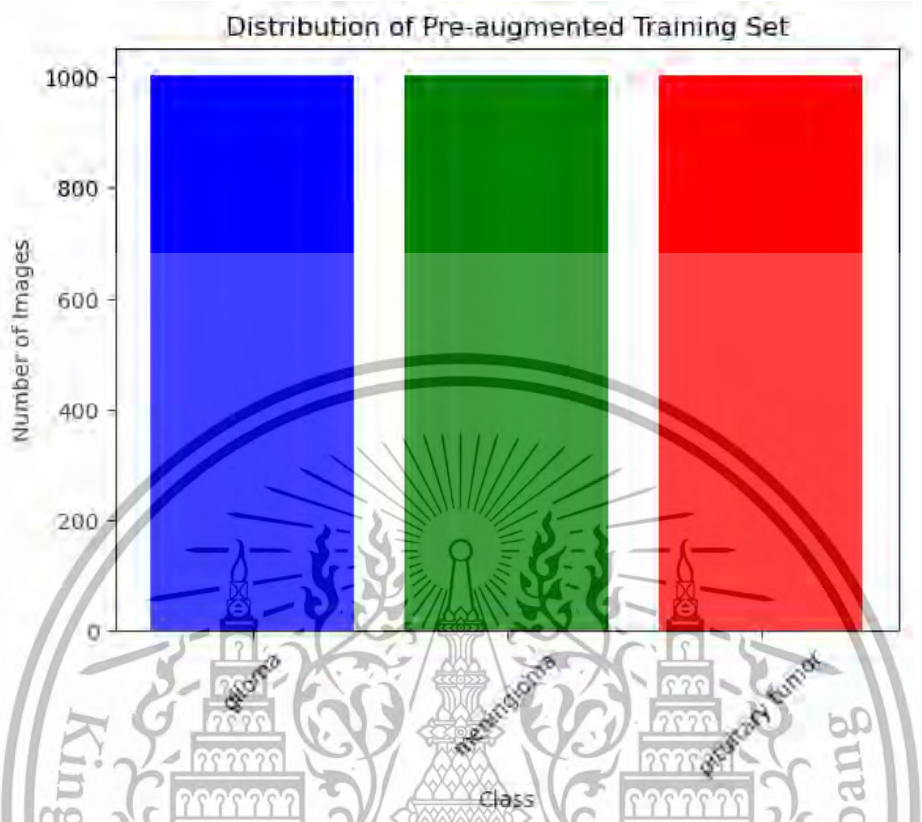

Figure 3.12 Pre-augmented training set with balanced classes

3.4.4 Computational cost

DL models in the field of images require computational power. To overcome these challenges while maintaining model efficiency we utilized Google Colab. By utilizing its GPU capabilities we were able to simplify the process of training our models without the need, for high performance hardware.

3.6 Summary

This chapter takes a dive into the processes and considerations involved in creating a DL model, for analyzing MRI images.

To begin with the chapter discusses where the data comes from. A brain tumor dataset obtained from Kaggle. This dataset contains types of brain tumors. Has been carefully preprocessed using techniques like normalization, median filtering and RGB conversion. The data is then divided into training, testing and validation sets to facilitate model development.

Next the chapter explores various model architectures to choose from. The models are based on DenseNet121, ResNet50V2 and InceptionResNetV2 frameworks. Each architecture brings its structure and approach to the table. All models are initialized with weights from ImageNet as a starting point for their efficiency.

The training of the model involves using a range of hyperparameters and strategies. These include adjusting epochs learning rates, weight decays and augmentation techniques in order to find the combination. Early stopping is also implemented to optimize efficiency.

The chapter also reflects on some challenges that were encountered during this process. These challenges include dealing with size handling diverse perspectives within MRI images addressing class imbalances, in representation and managing computational costs.

Every hurdle we faced throughout the process presented an opportunity for us to come up with groundbreaking solutions. Eventually we were able to propose a range of strategies that included techniques such, as data augmentation ensuring a representation of MRI views synthetic oversampling to tackle imbalanced classes and harnessing the power of Google Colab. These methods became our tools in the fight against the challenges we encountered. The entire journey of acquiring data, selecting models training them addressing challenges head, on and finding solutions provided us with an understanding of the dedication, innovation and strategic planning involved in developing our MRI image analysis model.

CHAPTER 4
EXPERIMENTAL RESULT AND DISCUSSION

4.1 Introduction

In the chapter 3 we discussed in detail of five fold cross validation , the design of training process to obtain multiple individual models, and creation of different network architectures. Moving forward to this chapter we will delve into the evaluation methods employed and the results that provide insights, into how each architecture performs in terms of effectiveness and class predictions.

The Chapter is organized as follow :

Section 4.1 : An overview of what to expect in this chapter, emphasizing the significance of evaluating our models and the methods chosen for this purpose.

Section 4.2 : Here, we present the outcomes of the 5-fold cross-validation for each architecture. We will discuss key performance metrics including accuracy, recall, and precision for each fold. Additionally, the mean accuracy and its standard deviation across the folds will be highlighted to gauge the model's robustness and performance variability.

Section 4.3: After our initial evaluations, this section introduces our experiment with ensemble techniques, specifically the 'Model Soups' approach. We will show the outcome of each model soup for different base model.

4.2 5-Fold Cross-Validation Results and Analysis

In this section we will explore the result and thorough analysis derived from the 5 Fold Validation process. We will present metrics of test accuracy and loss for each of the five folds, which will provide insights, into how the model performs across different data splits using the three different base models. Additionally we will highlight the Iteration that demonstrated performance among the five iterations delving into its classification report and accompanying confusion matrix for an understanding. Furthermore we will examine the mean accuracy and standard deviation metrics of our three base models. This comparison aims to shed light on the strengths and areas for improvement within the validation framework, for each model.

Test accuracy and loss of each Iteration of ResNet50V2 as base model		
Model on Iteration	Test accuracy	loss
Iteration 1	0.9522	0.1425
Iteration 2	0.9739	0.0904
Iteration 3	0.9435	0.2127
Iteration 4	0.9543	0.1172
Iteration 5	0.9609	0.1465

Table 4.1 Test accuracy and loss of each iteration of ResNet50V2 as base model

The provided table 4.1 shows the outcomes of a 5 Fold Cross Validation using the ResNet50V2 architecture as the model. From Fold1 to Fold5 represents a subset of the dataset for validation while the rest of the data is used for training. In terms of accuracy Iteration1 achieved a 95.22% indicating that it made predictions for over 95% of test instances in this particular subset. Among all Iteration, Iteration 2 stood out with the accuracy rate of 97.39% while iteration3, iteration4 and iteration5 had accuracies of 94.35%, 95.43% and 96.09% respectively. The loss metric, which measures how much error is present in model predictions revealed that iteration1 had a loss value of 0.1425 . Surprisingly in addition to having the accuracy rate iteration2 also had the loss value at just 0.0904 .On the hand when it comes to loss values alone it was observed that

iteration3 had the value at 0.2127 – suggesting a greater deviation between predicted and actual outcomes in this particular iteration. As for folds like iteration4 and iteration5 they exhibited losses of 0.1172 and 0.1465 respectively.

In summary all iteraion showcased levels of accuracy; however it was evident that iteration2 emerged as the performer among them all.

Test accuracy and loss of each iteration of DenseNet121 as base model		
Model on Iteration	Test accuracy	loss
Iteration1	0.9674	0.0775
Iteration 2	0.9674	0.0737
Iteration 3	0.9717	0.0805
Iteration 4	0.9609	0.1584
Iteration 5	0.9717	0.0982

Table 4.2 Test accuracy and loss of each iteration of DenseNet121 as base model

The table 4.2 Show the performance metrics of the base model during a 5 cross validation. When looking at accuracy both Iteration1 and iteration2 achieved the score of 96.74% while iteration3 and iteration5 performed better with a peak accuracy of 97.17%. On the hand iteration4 reported an accuracy of 96.09%. In terms of loss there was generally an inverse relationship, with accuracy. iteration2 had the loss value of 0.0737 while iteration4 had the highest at 0.1584. These results highlight the performance of the DenseNet121 architecture across different data partitions with slight variations, in its effectiveness.

Test accuracy and loss of each iteration of InceptionResNetV2 as base model		
Model on Iteration	Test accuracy	loss
Iteration 1	0.9652	0.1250
Iteration 2	0.9717	0.1237
Iteration 3	0.9543	0.1691
Iteration 4	0.9565	0.1367
Iteration 5	0.9609	0.1393

Table 4.3 Test accuracy and loss of each iteratrion of InceptioResNetV2 as base model

The table 4.3 presents the evaluation metrics obtained by using the InceptionResNetV2 as the model, in a 5 Cross Validation process. In terms of accuracy iteraion2 stands out with the score of 97.17% closely followed by iteration4 at 96.65%. The other folds also exhibit accuracies above the 96% threshold. Regarding the loss metric iteration3 shows the value at 0.1691 while iteration2 has the lowest, at 0.1237. These results highlight how consistently and diversely the InceptionResNetV2 model performs when assessed on parts of the data.

Classification Report for Iteration 2 (ResNet50V2)				
Class/Measure	Precision	Recall	F1-Score	Support
Glioma	0.97	0.99	0.98	190
Meningioma	0.97	0.94	0.95	128
Pituitary Tumor	0.98	0.98	0.98	142
Macro Avg	0.97	0.97	0.97	460
Weight Avg	0.97	0.97	0.97	460

Table 4.4 Classification report for Iteration2(ResNet50V2)

The table 4.4 shown presents a classification report, for The best iteration model (iteration2) using the ResNet50V2 model. When it comes to the Glioma class the model achieves a precision score of 0.97 and a recall score of 0.99 resulting in an F1 Score of 0.98. The Meningioma class has values with an F1 Score of 0.95 while the Pituitary Tumor class performs similarly to the Glioma class achieving an F1 Score of 0.98 as well. Taking into account all classes and averaging the metrics, both Macro and Weighted averages consistently show scores of 0.97 for precision, recall and F1 Score. This indicates that the model performs across all tumor classifications, in this iteration.

Classification Report for Iteration 3 (DenseNet121)				
Class/Measure	Precision	Recall	F1-Score	Support
Glioma	0.98	0.99	0.98	190
Meningioma	1.00	0.91	0.96	128
Pituitary Tumor	0.94	1.00	0.97	142
Macro Avg	0.97	0.97	0.97	460
Weight Avg	0.97	0.97	0.97	460

Table 4.5 Classification report for iteration 3 (DenseNet121)

Table 4.5 above show the classification report for iteration3, which was generated using the model. In this fold the Glioma class displays a precision of 0.98 and a recall rate of 0.99 resulting in an F1 Score of 0.98. Furthermore the Meningioma class achieves a precision score of 1.00 although its recall is slightly lower at 0.91 resulting in an F1 Score of 0.96. Regarding the Pituitary Tumor category the model showcases a precision value of 0.94 and a flawless recall rating of 1.00 leading to an F1 Score of 0.97. When considering metrics (both Macro and Weighted) all three measures—precision, recall and F1 Score— average at a value of 0.97 across different tumor types within this folds performance evaluation using the DenseNet121 model.

Classification Report for Iteration 2 (InceptionResNetV2)				
Class/Measure	Precision	Recall	F1-Score	Support
Glioma	0.98	0.99	0.99	190
Meningioma	0.99	0.91	0.95	128
Pituitary Tumor	0.95	0.99	0.97	142
Macro Avg	0.97	0.97	0.97	460
Weight Avg	0.97	0.97	0.97	460

Table 4.6 Classification report for iteration 2 (InceptionResNetV2)

The classification report 4.6 showcases the performance metrics for iteration 2, leveraging the InceptionResNetV2 model. For the Glioma class, the model achieves a precision of 0.98, a recall of 0.99, and an F1-Score of 0.99. The Meningioma class records a precision of 0.99, paired with a recall of 0.91, culminating in an F1-Score of

0.95. The Pituitary Tumor class posts a precision of 0.95 and a recall of 0.99, leading to an F1-Score of 0.97. Consistently, both the Macro and Weighted averages reflect values of 0.97 across the metrics. These results underline the InceptionResNetV2 model's effective classification capabilities in this fold.

Figure 4.1 Confusion Matrix of Model from Iteration 3 DenseNet121

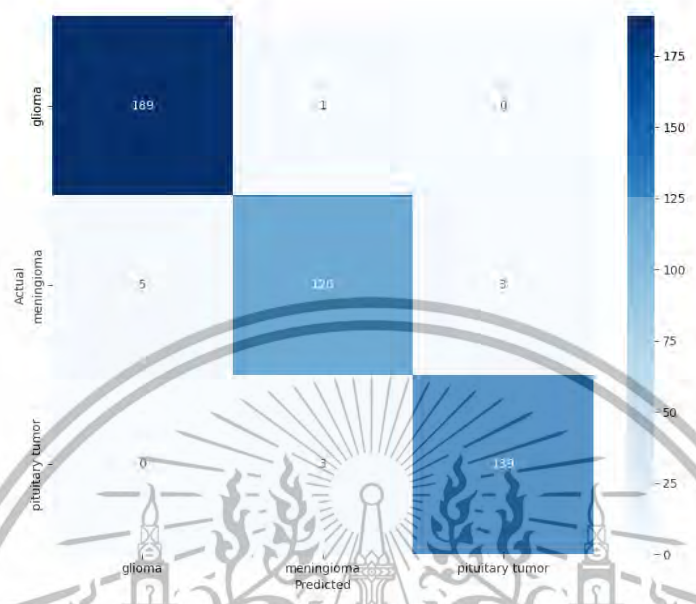

Figure 4.2 Confusion Matrix of model from Iteration 2 InceptionResNetV2

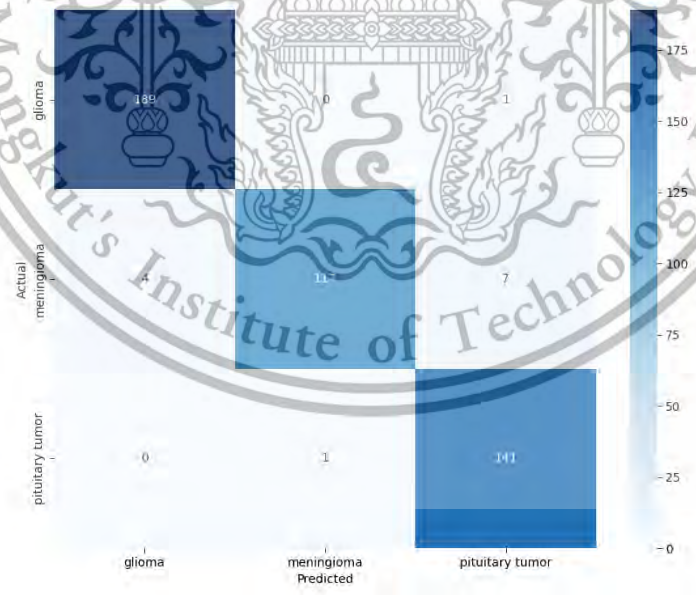

Figure 4.3 Confusion Matrix of model from Iteration 2 ResNet50V2

Mean accuracy and Standard deviation		
Model	Mean acuuracy	Standard deviation
InceptionResNetV2	0.9617	0.0062
ResNet50V2	0.9570	0.0101
DenseNet121	0.9678	0.0040

Table 4.7 Mean accuracy and Standard deviation

The table 4.7 above shows the average accuracy and variability of three models—InceptionResNetV2, ResNet50V2 and DenseNet121—evaluated on a test set across five different divisions. The average accuracy represents the performance of each model, across these divisions while the variability is a measure of how much the accuracies deviate from this average.

Based on the table DenseNet121 achieves the accuracy of 0.9678 followed closely by InceptionResNetV2 with 0.9617 and then ResNet50V2 with 0.9570. It is worth noting that while DenseNet121 has the accuracy it also exhibits the lowest variability with a standard deviation of 0.0040. On the hand even though ResNet50V2 has a lower mean accuracy it shows greater variation, in its results as indicated by its standard deviation of 0.0101. These insights are valuable as they not provide an indication of each models performance (average accuracy) but also give an understanding of how consistent or reliable their results are (standard deviation).

4.3 The Model Soups result

This section we will dive into the outcomes of test accuracy and loss across a range of models. We specifically examine 20 setups for each of our three chosen base architectures; DenseNet121, ResNet50V2 and InceptionResNetV2. Although these models have foundations they differ in terms of hyperparameters leading to performance results. To leverage the strength of these models we utilize techniques by combining the 20 models from each base architecture. This approach allows us to observe the combined effects and nuances, in performance when ensembling models with shared architecture but distinct hyperparameters. Subsequently we will present an analysis that compares the results obtained from our techniques against the performance

of the best performing individual model, for each base architecture. The goal is to identify the approach tailored to our dataset and objectives.

Hyperparameter	Possible Values		
Epochs	20	30	40
Learning Rate	1e-4	1e-5	5e-5
Weight Decay	1e-4	1e-5	5e-5
Augments	Augment	Augment1	None

Table 4.8 Hyperparameter values

```
all_combinations = list(product(epochs, learning_rate, weight_decay, augments))
random.shuffle(all_combinations)

parameters = [{"epochs": comb[0], "learning_rate": comb[1], "weight_decay": comb[2], "aug_map": comb[3], "aug_id": augment_ids.get(comb[3], 0)} for comb in all_combinations[:20]]
```

Figure 4.4 Random select function

During the training process we randomly selected hyperparameters, for each model from the provided ranges and sets. To diversify our models and understand how different hyperparameters affect performance when combined we trained them with data augmentation methods such as augment, augment1 or None which when the process select None , it will use the preaugmented balance classes training set instread of doing on-fly augmentation like Augment and Augment1 . This approach ensured a range of models with hyperparameters to analyze their collective impact, on performance.We also set an Early stopping to make sure the training process finished with the best weight. Eventually, We obtained 20 individual models with different hyperparameter.

Top 5 individual Models using ResNet50V2 as base Model							
Individual Model	Epochs	Actual epoch	Learning Rate	Weight Decay	Augmentation_type	Test ACC	Loss
1	30	7	1e-4	5e-5	None	0.967	0.109
2	40	10	5e-5	5e-5	Augment1	0.957	0.139
3	40	21	1e-5	1e-4	Augment	0.957	0.164
4	20	9	1e-4	1e-4	Augment1	0.956	0.121
5	30	14	5e-5	1e-4	Augment	0.954	0.168
……...	……...	……..	…….	……	…….	……..	…….

Table 4.9 Top 5 individual Models using ResNet50V2 as base Model

For the ResNet50V2 as based model

- The top-performing model had a test accuracy of 96.7% with a configuration Actual epoch at 7 , learning rate of 1e-4, weight decay of 5e-5, and None augmentation (preaugmented balance classes)

Top 5 individual Models using InceptionResNetV2 as base Model							
Individual Model	Epochs	Actual epoch	Learning Rate	Weight Decay	Augmentation_type	Test ACC	Loss
1	30	16	1e-4	5e-5	None	0.972	0.084
2	20	18	1e-4	1e-4	None	0.969	0.122
3	40	18	1e-4	1e-5	None	0.963	0.106
4	20	20	5e-5	5e-5	Augment	0.963	0.163
5	40	24	1e-5	5e-5	Augment	0.956	0.146
……...	……...	……..	…….	……	…….	……..	…….

Table 4.10 Top 5 individual Models using InceptionResNetV2 as base Model

For the InceptionResNetV2 base model

- The best model achieved an accuracy of 97.2% acutual epochs at 16, learning rate of 1e-4, aweight decay of 5e-5, and no data augmentation (preaugmented balance classes)

Top 5 individual Models using DenseNet121 as base Model							
Individual Model	Epochs	Actual epoch	Learning Rate	Weight Decay	Augmentation_type	Test ACC	Loss
1	30	23	1e-4	5e-5	Augment1	0.987	0.047
2	20	14	1e-4	5e-5	Augment1	0.978	0.074
3	30	20	5e-5	1e-5	Augment	0.974	0.080
4	40	15	5e-5	1e-4	Augment1	0.971	0.087
5	30	22	5e-5	1e-5	Augment	0.969	0.108
……...	……...	……..	……	……	……	……..	…….

Table 4.11 Top 5 individual Models using DenseNet121 as base Model

For the DenseNet121 as based model

- The best performance model has the test accuracy of 98.7% with a comnination of actual epoch at 23 , learning rate of 1e-4 , weight decay of 5e-5, and Augmented1

```python
def uniform_soup(model_paths, test_ds, model_fun, evaluate_fun):

    # Load models and their weights
    models = [model_fun() for _ in model_paths]
    for model, path in zip(models, model_paths):
        model.load_weights(path)

    # Uniform Soup Method: Average the weights of all models
    ensemble_weights = [m.get_weights() for m in models]
    avg_weights = np.mean(ensemble_weights, axis=0)

    ensemble_model = model_fun()
    ensemble_model.set_weights(avg_weights)

    # Evaluate ensemble model on test dataset
    loss, acc, class_report = evaluate_fun(ensemble_model, test_ds)

    return ensemble_model, loss, acc, class_report
def greedy_soup_v2(model_paths, test_ds, model_fun, evaluate_fun):

    # Load models and their weights
    models = [model_fun() for _ in model_paths]
    for model, path in zip(models, model_paths):
        model.load_weights(path)

    # Evaluate each model on validation/test dataset
    val_results = [evaluate_fun(m, test_ds)[1] for m in models]

    # Greedy Soup Method
    ranked_candidates = list(range(len(models)))
    ranked_candidates.sort(key=lambda x: -val_results[x])

    current_best = val_results[ranked_candidates[0]]
    best_ingredients = [ranked_candidates[0]]

    for i in tqdm(range(1, len(models))):
        ingredient_indices = best_ingredients + [ranked_candidates[i]]
        ensemble_weights = [models[j].get_weights() for j in ingredient_indices]
        avg_weights = np.mean(ensemble_weights, axis=0)
        ensemble_model = model_fun()
        ensemble_model.set_weights(avg_weights)
        current_accuracy = evaluate_fun(ensemble_model, test_ds)[1]

        if current_accuracy > current_best:
            current_best = current_accuracy
            best_ingredients.append(ranked_candidates[i])

    # Get final ensemble model with best ingredients
    final_weights = np.mean([models[i].get_weights() for i in best_ingredients], axis=0)
    final_model = model_fun()
    final_model.set_weights(final_weights)
    loss, acc, class_report = evaluate_fun(final_model, test_ds)
    return final_model, loss, acc, class_report
```

Figure 4.5 Uniform Soup and Greedy Soup function

Once we collected 20 models, for each base model we utilized techniques known as the Uniform Soup and the Greedy Soup as shown in the Figure 4.5. These methods were applied individually to each base model resulting in Uniform Soup and Greedy Soup models, for every base model architecture.

Model	Test Accuracy	Loss
Best individual model (ResNet50V2)	0.967	0.109
Uniform soup Model (ResNet50V2)	0.631	1.09
Greedy soup Model (ResNet50V2)	0.967	0.109
Best individual model (InceptionResNetV2)	0.972	0.084
Uniform soup Model (InceptionResNetV2)	0.732	1.07
Greedy soup Model (InceptionResNetV2)	0.972	0.084
Best individual model (DenseNet121)	0.987	0.047
Uniform soup Model (DenseNet121)	0.441	0.882
Greedy soup Model (DenseNet121)	0.987	0.047

Table 4.12 Compartion of best individual model , greedy soup , Uniform soup

The table 4.12 presents a comparison of the best individual model, uniform soup models and greedy soup models, across three different base architectures; ResNet50V2 InceptionResNetV2 and DenseNet121. Interestingly the greedy soup model yielded results to the individual model indicating that they utilized the same model weights. However it appears that the uniform soup models did not perform well across all base models.

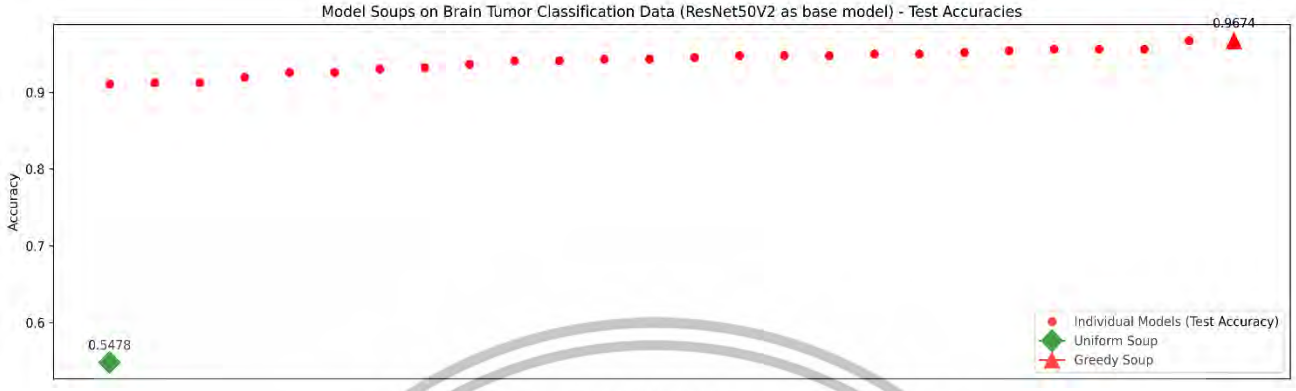

Figure 4.6 Test accuracy of each model using ResNet50V2 as a based model

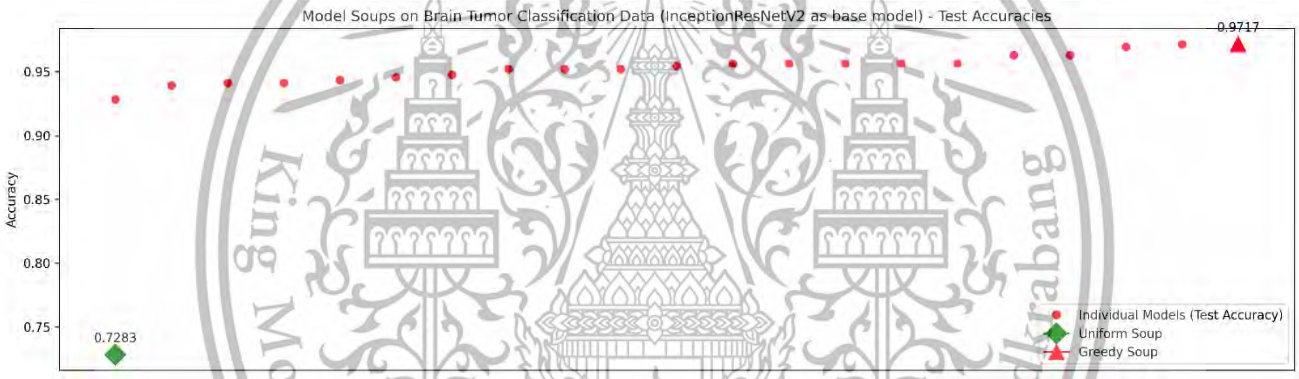

Figure 4.7 Test accuracy of each model using InceptionResNetV2 as a based model

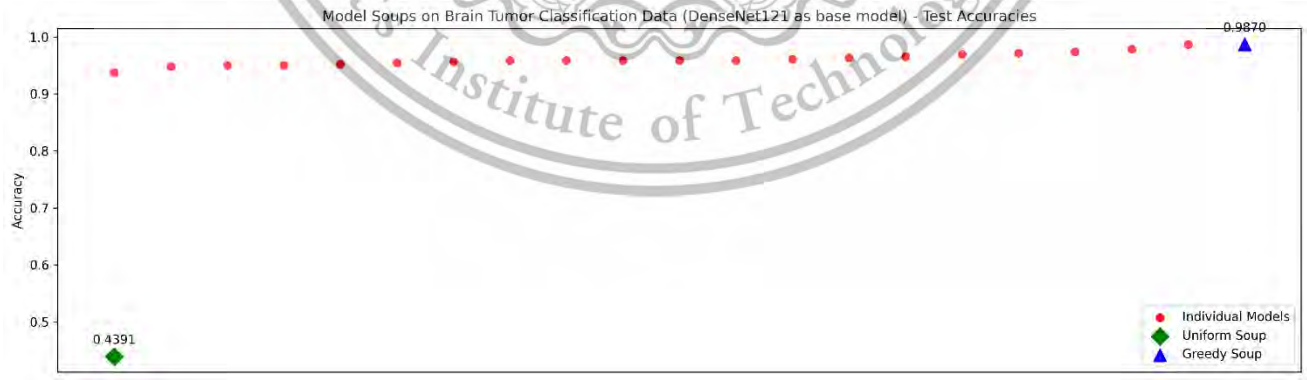

Figure 4.8 Test accuracy of each model using DenseNet121 as a based model

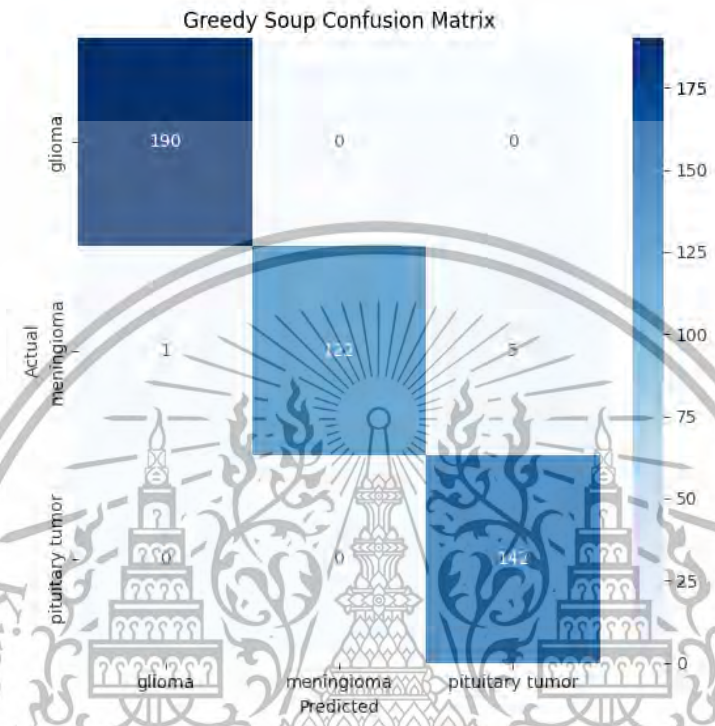

Figure 4.9 Confusion Matrix of greedy soup model using Dense121 as a based model

Confusion Matrix Analysis on greedy soup using DenseNet121 as base Model

- For the Glioma class, the model showcased perfect classification which it completely predict correctly for all classes
- Meningioma classification was nearly perfect which it correctly classify 122 samples. However, there was 1 sample misclassify as Glioma and 5 were predicted as Pituitary Tumor.
- The Pituitary Tumor class show accurate classification for all 142 samples without any errors.

Classification Report for Greedy soup (DenseNet121)				
Class/Measure	Precision	Recall	F1-Score	Support
Glioma	0.99	1.00	1.00	190
Meningioma	1.00	0.95	0.98	128
Pituitary Tumor	0.97	1.00	0.98	142
Macro Avg	0.99	0.98	0.99	460
Weight Avg	0.99	0.99	0.99	460

Table 4.13 Classification Report for Greedy soup (DenseNet121)

Classification Report in table 4.13 Insights of Greedy soup using DenseNet121 as base model:

- Glioma achieved a classification, with precision, recall and F1 Score of 0.99, 1.00 and 1.00 respectively.
- Meningioma demonstrated the highest precision of 1.00 and a recall of 0.95 resulting in an F1 Score of 0.98. The slight decrease in recall accounts for the mentioned misclassifications.
- Pituitary Tumors performance was strong with both precision and recall at 0.97 resulting in an F1 Score of 0.98.
- The models overall performance (Macro and Weighted Average) is remarkable as it approaches values close, to 1 indicating reliability and effectiveness.

4.4 Web application result

Finally we've implemented our model on a web application using Streamlit. By integrating our trained model with Streamlit users can effortlessly upload MRI images. Instantly receive brain tumor classifications directly on the web interface. This not

offers a user platform, for end users but also demonstrates how practical and useful our model is, in real world scenarios. Through Streamlit we transform our model into an accessible tool that enables quick and accurate diagnosis of brain tumors.

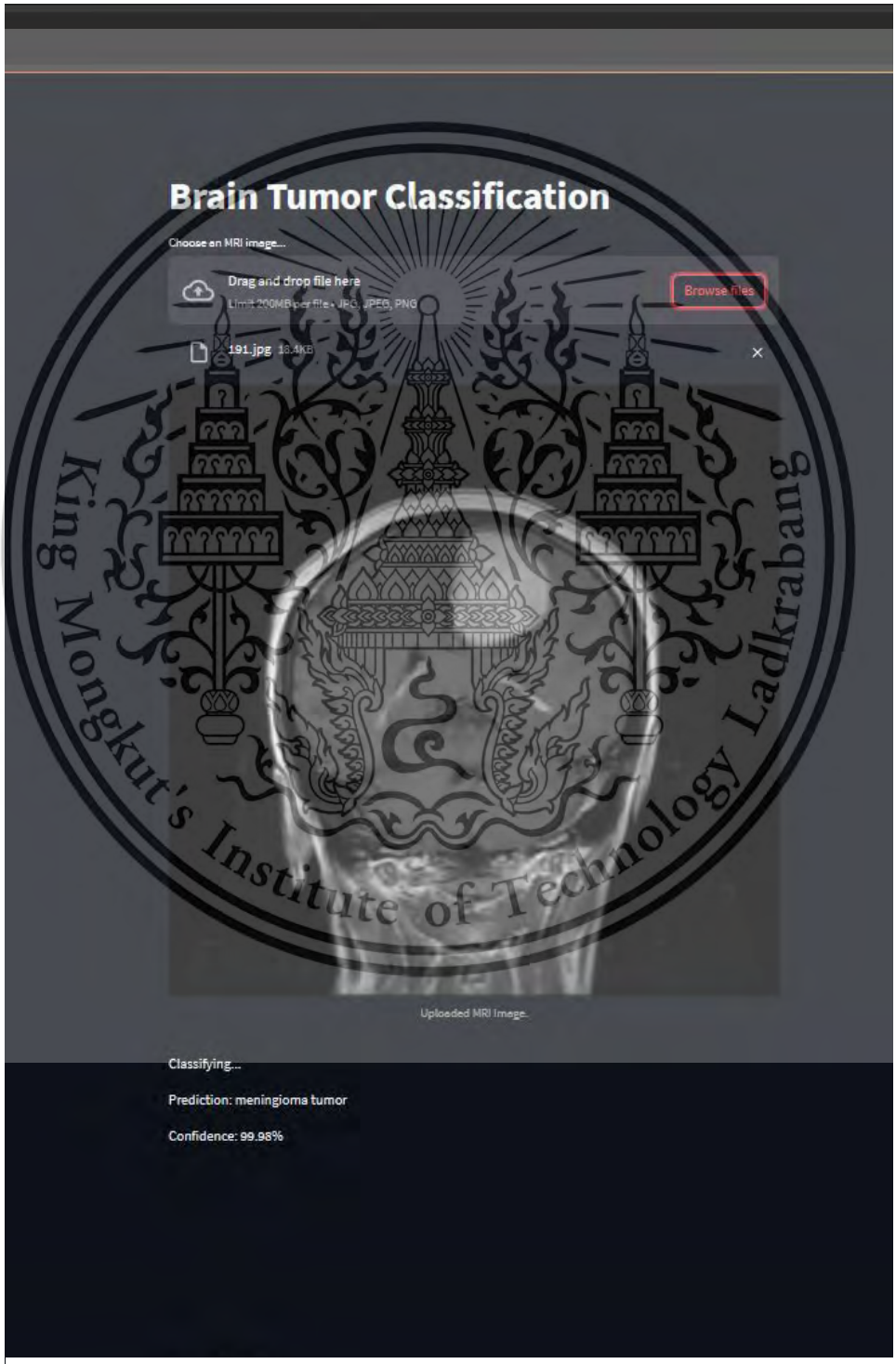

Figure 4.10 Web application of brain tumor classificatiion

CHAPTER 5
CONCLUSION

5.1 Introduction

In this Chapter, we first summarize the work described in this report in Chapter 5.2. Then we draw several conclusions about important parts of the work undertaken in Chapter 5.3, and finally, in Chapter 5.4 we discuss future work and how we see other technologies helping support projects such as this one.

5.2 Summary

Chapter 1 introduced the importance of brain tumors classification using DLhelp to the medical healthcare.

Chapter reviewed the state-of-the-art in MRI images, and network architectures. Different models were introduced and described. This chapter also review the theory of five-fold cross validation , model soups , and evaluation matric

Chapter 3 describes the design of developing deep learning model. The separate functions of model architechture that support the requirements were then described in more detail, including training process, and comparing the composite architecture models.

Chapter 4 described and compare the result of each models using key evaluation matric and finally show the implementation of the best performance model (greedy soup model) with web application of brain tumor classification

Chapter 5 conclude the important information about this project and suggestion any of the future work of this project.

5.3 Conclusions

This project was initiated with the aim of improving the accuracy of learning models used for classifying three types of brain tumors. To achieve this we utilized various type of techniques such as five fold cross validation and model soup. Our main objective was not to develop a performing model but also to integrate it into web application. This way we wanted to bridge the gap, between machine learning solutions and practical clinical applications.

Our approach involved a process of refining models to determine the optimal configuration. Unfortunely, we discovered that the 'uniform_soup' model despite its foundation did not meet our performance expectations when compared with other models. This highlighted the complexity of model selection, where combining techniques and averaging model weight does not always result in effectiveness. On the hand the 'greedy_soup' model have the similar evaluation result as the best individual model. The conclude that greedy soup model is selected only the best performance model weight.

We relied on five cross validation as a essential tool, for assessing and validating our models.Through an approach of dividing the dataset and rotating the validation set across sections we ensured not only the reliability of our evaluation but also minimized the risks of overfitting. This rigorous methodology allowed us to interpret our models performance, with confidence and reliability reinforcing the credibility of our obtained results.

To conclude our investigation supports the hypothesis presented in Chapter 1 regarding the effectiveness of methods in brain tumor classification tasks. Our test accuracy of the best model either the best individual model or greedy soup model using DenseNet121 as feature extractor has remarkable result which is 98.7%. Although the model soup method do not improve in accuracy, we recived the remarkable result of the test accury of individual model.

5.4 Future Scope

In our project we created an ensemble of 20 models, for our model soup essemble technique.With this number of models , it do not improved performance compare with the best individual model. The future of this project is to trained more model to received as much of diverse hyperparameter set of models which could be further increase in accuracy using model soups essemble technique.

In term of web application, One important factor that could additionally be integrated into the web application is a database system specifically created to gather and save user information. This collection of data can be utilized for ongoing training and improvement. By continuously updating and expanding the dataset, the model can become more robust and effective over time.

REFERENCES

[1] A. S. Ahuja, "The impact of artificial intelligence in medicine on the future role of the physician," *PeerJ*, vol. 7, p. e7702, Oct. 2019, doi: 10.7717/peerj.7702.

[2] Z. Rasheed *et al.*, "Brain Tumor Classification from MRI Using Image Enhancement and Convolutional Neural Network Techniques," *Brain Sci.*, vol. 13, no. 9, p. 1320, Sep. 2023, doi: 10.3390/brainsci13091320.

[3] C. Gungen, O. Polat, and R. Karakis, "Classification of Brain Tumors using Convolutional Neural Network from MR Images," in *2020 28th Signal Processing and Communications Applications Conference (SIU)*, Gaziantep, Turkey: IEEE, Oct. 2020, pp. 1–4. doi: 10.1109/SIU49456.2020.9302090.

[4] P. Ratan, "What is the Convolutional Neural Network Architecture?," Analytics Vidhya. Accessed: Nov. 04, 2023. [Online]. Available: https://www.analyticsvidhya.com/blog/2020/10/what-is-the-convolutional-neural-network-architecture/

[5] C. Iorga and V.-E. Neagoe, "A Deep CNN Approach with Transfer Learning for Image Recognition," in *2019 11th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, Pitesti, Romania: IEEE, Jun. 2019, pp. 1–6. doi: 10.1109/ECAI46879.2019.9042173.

[6] "ML Pipeline: Highlights the Challenge of Manual Feature Extraction," linuxtut.com. Accessed: Nov. 04, 2023. [Online]. Available: https://www.linuxtut.com/en/3d5b86ea5cbe75c8ab89/

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Identity Mappings in Deep Residual Networks." arXiv, Jul. 25, 2016. doi: 10.48550/arXiv.1603.05027.

[8] E. U. H. Qazi, T. Zia, and A. Almorjan, "Deep Learning-Based Digital Image Forgery Detection System," *Appl. Sci.*, vol. 12, no. 6, Art. no. 6, Jan. 2022, doi: 10.3390/app12062851.

[9] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning," *Proc. AAAI Conf. Artif. Intell.*, vol. 31, no. 1, Feb. 2017, doi: 10.1609/aaai.v31i1.11231.

[10] N. Bhat, K. V. Archana Hebbar, S. Bhat, Jayalakshmi, Pooja, and D. N. Harshitha, "Multilabel Spatial Image Recognition using Deep Convolutional Neural Network," in *2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, Coimbatore, India: IEEE, Nov. 2020, pp. 1–6. doi: 10.1109/ICECA49313.2020.9297545.

[11] G. Jee, H. Gm, M. K. Gourisaria, V. Singh, S. S. Rautaray, and M. Pandey, "Efficacy Determination of Various Base Networks in Single Shot Detector for Automatic Mask Localisation in a Post COVID Setup," *J. Exp. Theor. Artif. Intell.*, vol. 35, no. 3, pp. 345–364, Apr. 2023, doi: 10.1080/0952813X.2021.1960638.

[12] M. Wortsman *et al.*, "Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time," in *Proceedings of the 39th International Conference on Machine Learning*, PMLR, Jun. 2022, pp. 23965–23998. Accessed: Nov. 04, 2023. [Online]. Available: https://proceedings.mlr.press/v162/wortsman22a.html

[13] V. García, J. S. Sánchez, and A. I. Marqués, "Synergetic Application of Multi-Criteria Decision-Making Models to Credit Granting Decision Problems," *Appl. Sci.*, vol. 9, no. 23, p. 5052, Nov. 2019, doi: 10.3390/app9235052.

[14] D. A. Korevaar, G. Gopalakrishna, J. F. Cohen, and P. M. Bossuyt, "Targeted test evaluation: a framework for designing diagnostic accuracy studies with clear study hypotheses," *Diagn. Progn. Res.*, vol. 3, no. 1, Art. no. 1, Dec. 2019, doi: 10.1186/s41512-019-0069-2.

[15] W. Sonarra, N. Vongmanee, N. Wanluk, C. Pintavirooj, and S. Visitsattapongse, "Detection and Classification of COVID-19 Chest X-rays by the Deep Learning Technique," in *2022 14th Biomedical Engineering International Conference (BMEiCON)*, Songkhla, Thailand: IEEE, Nov. 2022, pp. 1–5. doi: 10.1109/BMEiCON56653.2022.10012094.

[16] jesse_jcharis, "Streamlit Projects – An App Challenge Series," JCharisTech. Accessed: Nov. 04, 2023. [Online]. Available: https://blog.jcharistech.com/2021/11/28/streamlit-projects-an-app-challenge-series/

[17] H. H. Sultan, N. M. Salem, and W. Al-Atabany, "Multi-Classification of Brain Tumor Images Using Deep Neural Network," *IEEE Access*, vol. 7, pp. 69215–69225, 2019, doi: 10.1109/ACCESS.2019.2919122.

[18] A. U. Haq, J. P. Li, S. Khan, M. A. Alshara, R. M. Alotaibi, and C. Mawuli, "DACBT: deep learning approach for classification of brain tumors using MRI data in IoT healthcare environment," *Sci. Rep.*, vol. 12, no. 1, p. 15331, Sep. 2022, doi: 10.1038/s41598-022-19465-1.

[19] J. Cheng *et al.*, "Enhanced Performance of Brain Tumor Classification via Tumor Region Augmentation and Partition," *PLOS ONE*, vol. 10, no. 10, p. e0140381, Oct. 2015, doi: 10.1371/journal.pone.0140381.

[20] J. Cheng *et al.*, "Retrieval of Brain Tumors by Adaptive Spatial Pooling and Fisher Vector Representation," *PLOS ONE*, vol. 11, no. 6, p. e0157112, Jun. 2016, doi: 10.1371/journal.pone.0157112.