PEOPLE’S DEMOCRATIC REPUBLIC OF ALGERIA

MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

UNIVERSITY KASDI MERBAH OF OUARGLA

Faculty of New Information and Communication Technologies

Department of Computer Science and Information Technology

MASTER THESIS

Domain: Computer science and Information technology

Field: Computer science

Specialty: Artificial Intelligence and Data Science

By: Bendob Khaled And Osmani Noufel Abdelouahab

THEME:

Lightweight CNN for Fast and Accurate CT/MRI brain Image Classification

EVALUATION DATE: 12/06/2025

JURY MEMBERS:

DR. Hamrouni Besma	JURY CHAIR
DR. MOHAMED-LAMINE BENDAOUD	SUPERVISOR
DR. Benchabana Ayoub	EXAMINER

Academic year: 2024 /2025

Acknowledgements

First and before everything, our utmost thanks go to Allah—first and foremost—who granted us the patience and ability to complete this achievement.

All praise is due to Allah, by whose grace good deeds are completed, and through whose guidance goals are achieved.

we would like to extend deep thanks and sincere gratitude to our supervising professor, **Mr. MOHAMED-LAMINE BENDAOUD**, whose advice, guidance, and continuous follow-up had a great impact on bringing this work to light. He generously shared his knowledge and experience, and was a constant source of support throughout the preparation of this research.

also we wish to express our heartfelt appreciation to **Mr. Aiadi OSSAMA**, who guided me through the early stages and played an important role in building a part of this work. May Allah reward him on my behalf and bless his efforts.

Special thanks go as well to all the professors of the Department of Media and Communication Sciences, who contributed to enriching my academic journey with their knowledge and support. You all have my deep respect and appreciation.

Dedication

وَقُل رَّبِّ ارْحَمْهُمَا كَمَا رَبَّيَانِي صَغِيرًا (24) الاسراء

“My Lord, have mercy upon them as they brought me up [when I was] small.” (24) AL-Isra

All praise is due to Allah, by whose grace good deeds are completed, whose guidance leads the lost to clarity, and through whose favor the striving soul reaches its goal.

To my beloved father – may Allah prolong your life and bless your health and well-being –
You are the one whose prayers sustained me, whose patience inspired me, and whose hard work set the example I follow. I kiss the hands that gave so much for me. I pray that Allah rewards you abundantly and makes me a source of pride and joy for you in this life and the next. Without your support and guidance, I would not have taken these steps.

To my dear mother – the crown on my head and the light of my heart –
You stayed awake for me, prayed for me in your sujood (prostration), and wiped away my weariness with your words and tenderness. I ask Allah to grant you paradise without reckoning and protect you from every harm. Your blessed prayers were the true key to this achievement. May Allah reward you for all you’ve done for me.

And to my brothers and sisters,
Who have shared the journey with me, and who were, after Allah, a support and a refuge—I pray that Allah preserves you, rewards you for every act of kindness, every encouraging word, and every sincere prayer. You complete the meaning of family, and with you, the path feels lighter.

I ask Allah to keep you all safe, to preserve the love and unity between us, and to make this work sincerely for His sake, and a source of pride and goodness for all of you.

BENDOB KHALED

شُكْرٌ وَعِرْفَان

بسم الله الرحمن الرحيم

"الحَمْدُ لِلَّهِ الَّذِي هَدَانَا لِهَٰذَا وَمَا كُنَّا لِنَهْتَدِيَ لَوْلَا أَنْ هَدَانَا اللَّهُ"

الحمد لله الذي بنعمته تتم الصالحات، حمداً يملأ الميزان، ويليق بجلال وجهه وعظيم سلطانه. وما هذا الجهد إلا قبسٌ من نوره، وفيضٌ من كرمه، وتوفيقٌ منه لإتمام عملٍ ما كان ليبلغ تمامه لولا عونه ومدده.

إلى من منحني اسمه لأحمله فخراً

إلى أبي، يا صاحب الظل الممتد الذي استظللت به من عثرات الطريق، ويا من علمتني بصمتك المهيب أن الأفعال أبلغ من كل كلام.

وإلى من نسجت من دعواتها حرزاً يحيط بي، ومن حنانها رداءً يأوي إليه تعبي...

إلى أمي، يا نبع الحياة الذي لا ينضب، ويا سر الطمأنينة في قلبي. هذا النجاح ليس سوى صدىً لدعوةٍ صادقةٍ همستِ بها من قيامك في جوف ليل.

إليكما معاً... يا من كنتما مبتدأ الحكاية ومنتهاها، أهدي هذه الثمرة التي سقيت بذورها بصبر السنين وخلاصة من عمريكما

إلى من يشاركونني لحن العمر وأجمل الذكريات...

إخوتي وأخواتي، كنتم السند الذي لا يميل، والضحكة التي تعيد الروح للروح

يصحب شيخا عارف المسالك يقيه في طريقه المهالك

الى شيخ المربي الطالب محمد الأزهر بن عبد الحفيظ بوغابة، شيخي الذي أستقي من بحر أدبه **وعلمه وقرآنه وأسراره**، فجزاه الله عنا كل **خير**، **وإلى** رفقة الصفوة من أهل القرآن، تلك الكوكبة النيّرة التي كانت مجالسها واحة **وسكينة وطمأنينة**

إلى أساتذتي الأجلاء، صناع الفكر وبناة الأجيال، أخص بالذكر والامتنان العميق أستاذي ومشرفي، (محمد الامين بن داود)، الذي كان لنا مرشداً حكيماً **ومعينا**، **والى** أستاذنا الفاضل (أسامة **عيادي) الذي** كان ولا زال يرافقنا بتوجيهاته **وتوصياته** في **هذه** الرحلة **الجامعية**، فبارك الله في علمكما **ونفع** بكما الأمة

إلى العقول التي جاورت عقلي،

إلى رفاق الدرب وزملاء **التخصص**، **الذين** تبارينَا فكراً وتشاركنا هَمّاً. كنتم خير **معين**، جمعنا الله في المعالي ما توالت الأيام

وأخيرا الى صديقي الصدوق

زميلي في **البحث**، خير ما أختم به شكري **وعرفاني**، الطالب خالد بن **الضب**، بارك الله في أم أنجبتك **وأب** رباك **ورعاك**.

أهدي هذا الجهد المتواضع إلى كل هؤلاء الأنفس الكريمة

دمتم جميعا في امان الله **وحفظه** و السلام عليكم

اللهم صلِّ وسلم على سيدنا محمد

عصماني نوفل عبد الوهاب

Abstract

Accurate and rapid diagnosis of brain tumors is crucial for improving patient outcomes. However, traditional methods rely on manual examination of MRI and CT images, a process that is time-consuming and labor-intensive and whose accuracy relies heavily on the expertise of specialists, which can lead to variability in results.

Technologies such as machine learning and deep learning have emerged as promising alternatives to automate this process. However, current models often require significant computational resources, limiting their use in immediate diagnosis or their generalization to end-user devices. This presents a challenge for real-world deployment.

This limitation has created a competitive environment for modifying existing models or developing effective new ones. To address this, our research aims to develop a lightweight convolutional neural network capable of balancing accuracy and efficiency, enabling faster and easier classification of brain images.

We used techniques such as image resizing and data augmentation to provide a suitable training environment for our model. We ensured our model's ability to generalize and provide excellent performance with both visible and unseen data, using techniques such as [Dropout] and Cross-validation. Finally, we compared the model's performance with pre-trained models [VGG16] and [ResNet50]. Experimental results showed that our model achieved an accuracy of **98%** a sensitivity of **98%.**

Keywords: brain tumors, MRI/CT images, deep learning, CNN, data augmentation

ملخص

يُعد التشخيص الدقيق والسريع لأورام الدماغ أمرًا بالغ الأهمية لتحسين نتائج المرضى. ومع ذلك، تعتمد الطرق التقليدية على الفحص اليدوي لصور الرنين المغناطيسي (MRI) والتصوير المقطعي المحوسب(CT) ، وهي عملية تستغرق وقتًا طويلاً وتتطلب جهدًا كبيرًا، كما أن دقتها تعتمد بشكل كبير على خبرة المتخصصين، مما قد يؤدي إلى تباين في النتائج.

ظهرت تقنيات مثل التعلم الآلي والتعلم العميق كبدائل واعدة لأتمتة هذه العملية. ومع ذلك، غالبًا ما تتطلب النماذج الحالية موارد حوسبة كبيرة، مما يحد من استخدامها في التشخيص الفوري أو تعميمها على الأجهزة الطرفية. وهذا يمثل تحديًا أمام تطبيقها في الواقع العملي.

أدى هذا القيد إلى خلق بيئة تنافسية لتعديل النماذج الحالية أو تطوير نماذج جديدة فعالة. وللتغلب على هذا التحدي، تهدف أبحاثنا إلى تطوير شبكة عصبية التفافية خفيفة الوزن قادرة على الموازنة بين الدقة والكفاءة، مما يتيح تصنيفًا أسرع وأسهل لصور الدماغ.

استخدمنا تقنيات مثل تغيير حجم الصور وتعزيز البيانات (Data Augmentation) لتوفير بيئة تدريب مناسبة لنموذجنا. كما ضمّنا آليات مثل الإسقاط العشوائي Dropout وCross-validation لضمان قدرة النموذج على التعميم وتقديم أداء ممتاز مع البيانات المرئية وغير المرئية. وأخيرًا، قارنّا أداء النموذج مع نماذج مُسبقة التدريب مثل [VGG16] و [ResNet50]. وأظهرت النتائج التجريبية أن نموذجنا حقق دقة قدرها 98%، وحساسية قدرها 98%.

الكلمات المفتاحية :أورام الدماغ، صورMRI/CT ، التعلم العميق، الشبكات العصبية الالتفافية، تعزيز البيانات.

Table of content

Table of figures

Table

General Introduction

Diagnosing brain tumors has been one of the most challenging disorders of the past two decades. As one study showed, they are one of the top ten causes of cancer death. They remain a serious health threat and are largely incurable. In the United States, the annual incidence of brain cancer is approximately 15-20 cases per 100,000 people. The 5-year survival rate for brain cancer patients has remained stable at approximately 75% over the past decade. [1] This rate will continue to improve in the coming years thanks to the efforts of the scientific community in early diagnosis of brain tumors.

Early and accurate diagnosis of the disease is crucial for both patients and physicians, as it helps improve patient outcomes through faster treatment planning and reduces the risk of disease progression and metastasis [2] Conversely, accurate detection enables physicians to make informed decisions about the extent of surgical resection, which can reduce complications and improve patient recovery **,** It also helps guide treatment strategies, allowing for personalized care by predicting the likelihood of tumor recurrence and survival rates

Brain tumors are typically diagnosed after the onset of symptoms including headache, nausea, personality changes, seizures, or focal neurological deficits. Brain tumors include gliomas, meningiomas, and pituitary tumors**.** [1]

These symptoms are monitored using medical imaging, which also plays a pivotal role in modern healthcare, enabling physicians

to visualize internal structures and processes for accurate diagnosis, treatment planning, and disease monitoring.

The most common types of medical imaging used are magnetic resonance imaging (MRI) and computed tomography (CT).

First, magnetic resonance imaging (MRI) is an effective technique widely used in detecting brain tumors due to its ability to differentiate between structures and tissues based on contrast levels [3]. It is also an advanced medical imaging technique that provides valuable information about the anatomy of human soft tissues [4]

Second, computed tomography (CT) is essentially a series of X-rays stitched together to create a three-dimensional image. These are often used to detect diseases because they do not take much time to complete and clearly show the boundaries between different body parts [5]

- In some cases, combining CT and MRI may provide better diagnostic information.

The process of detecting tumors and extracting tumor regions from brain MRIs is time-consuming and requires significant effort from clinical experts or radiologists, while accuracy depends solely on their experience. This has led to an urgent need for more advanced and automated methods to detect and classify brain tumors quickly and accurately.

In recent years, various studies have been conducted on brain tumor classification. Deep learning techniques, particularly convolutional neural network (CNN) models, have shown promising results.

CNNs offer a promising approach by leveraging their ability to automatically learn and extract features from images, eliminating the need for manual extraction.

This automatic feature learning enables CNNs to capture complex patterns and subtle image characteristics that might be overlooked by human observers or traditional algorithms.

Moreover, CNNs are highly adaptable and can be trained to perform a variety of tasks, including tumor diagnosis, classification, and determining their size and location, making them ideal for brain tumor detection and diagnosis. Studies indicate that some models achieve accuracy between 90% and 99% (as we will see in previous studies).

However, training deep convolutional neural networks typically requires large amounts of labeled data, which can be challenging in the field of medical imaging due to data scarcity and privacy concerns.

These limitations have prompted the adoption of other advanced techniques such as transfer learning, whose main advantage is that it leverages the knowledge gained from models previously trained on large, multi-purpose datasets (such as ImageNet) and applies it to a new, relevant task with limited data.

This approach offers several solutions to the problem of limited medical data and available computational resources, making deep learning easier and more effective for analyzing medical images.

the contributions of this thesis can be summarized as follows:

- Development of a CNN architecture specifically tailored for brain tumor classification from MRI/CT images.
- Evaluation of the effectiveness of transfer learning in improving the performance of the CNN model with limited data.
- A comparison of the performance of the proposed CNN with transfer learning against

the remainder of this thesis is structured as follows:

- Chapter 1: Literature Review: This chapter provides a background, review of existing literature on brain tumor imaging, deep learning methods for medical image analysis, and transfer learning approaches.
- Chapter 2: Methodology: This chapter details the proposed CNN architecture, the transfer learning techniques employed, the dataset used for training and evaluation, and the experimental setup.
- Chapter 3: Results and Discussion: This chapter presents the experimental results, compares the performance of the proposed method with existing techniques, and discusses the implications of the findings.
- Chapter 4: Conclusion: This chapter summarizes the key findings of the thesis, highlights the contributions of the work, and suggests directions for future research

Chapter 1

1- Literature Review

Today’s rapid technological advances have led to a boom in new methods and tools for analyzing information. Many existing technologies have been significantly improved by in corporations artificial intelligence, particularly machine learning. Machine learning, a data mining powerhouse, excels at sifting through this massive pool and producing meaningful insights. As one of the fastest evolving areas of data science, the potential applications of machine learning seem limitless. ML primarily focuses on enabling machine programs to ac quire knowledge through experience and perform functions autonomously without relying on external oversight in the form of rule-based programming. The overall goal of ML is to improve the ability of machines to fix errors without specific pre-programmed instructions. Automated thinking becomes essential to derive insights from the vast repositories of big data, whether structured or unstructured. Machine learning helps predict patterns based on stored data, facilitating informed decisions and analysis. In this chapter we present the main background of our work, including details on machine learning and computer vision. Also, image processing and an overview of deep learning.

1.1 Artificial intelligence

Artificial intelligence (AI) is a branch of computer science and technology that aims to develop theories, methods, algorithms, and applications to simulate and augment human intelligence. The current paradigm shift in AI research involves a shift from a rule-based approach, where human intervention dictates problem-solving procedures, to a more autonomous problem-solving paradigm. In this new world, AI systems are equipped with algorithms that enable them to learn independently and develop solutions to specific problems.[6]

1.2 Machine learning

Machine learning is the science (and art) of programming computers to enable them to learn from data. It focuses on teaching computers how to learn without having to program them for specific tasks. The basic idea behind machine learning is the ability to create algorithms that learn from and predict data. Machine learning can be classified into several categories based on the nature of the learning data and the interaction between the learner and the environment. [7]

1.2.1 Types of Machine Learning

Machine learning can be divided into two main parts are **Supervised Learning** and **Unsupervised Learning**, and another combines the first two types called **Semi Supervise Learning**, In addition to **Reinforcement Learning**

1. **Supervised learning:** is typically the task of machine learning to learn a function that maps an input to an output based on sample input-output pairs. It is also called a task-driven approach. The most common supervised tasks are **classification** and **regression**. [7]

   Classification: is the process of organizing a dataset into different classes or categories. This involves learning from the data set and assigning labels or categories to new data points. These classes, also called labels or targets, can be applied to both linear and nonlinear data. Classification is often used when the output has finite and discrete values and the training classes can be expressed as binary choices, **e.g**. B. Yes or no, 0 or 1. [8]

Regression is a type of machine learning task that involves predicting a continuous value. Regression tasks are typically solved using supervised learning algorithms that are trained on a dataset of input-output pairs, where the inputs are the features of the data and the outputs are the target values.

Examples of regression scenarios include:

- Predicting house prices based on house attributes such as the number of bedrooms, location, or size.
- Predicting future stock prices based on historical data and current market trends.
- Predicting sales of a product based on advertising budgets. [7]

2. **Unsupervised learning:** is where the input data is unlabeled and the system tries to learn structure from that data automatically, without any human guidance. It is also called the data-driven approach. The most common unsupervised tasks are **clustering** and **dimensionality reduction** [7]

Clustering:

The practice of finding naturally occurring groups or clusters within multidimensional data using a similarity metric (such as Euclidean distance) is known as data clustering. It is a crucial step in machine learning and pattern recognition. Additionally, data clustering is a vital function. In Artificial Intelligence (AI) clustering algorithms are used in many applications, such as: image segmentation, vector and color image quantization, data mining, compression, machine learning, etc [9]

Examples of clustering scenarios include:

- Understanding segments of hotel guests based on habits and characteristics of hotel choices.
- Identifying customer segments and demographics to help build targeted advertising campaigns

Dimension reduction:

One essential tool for evaluating and interpreting high dimensional data is the application of dimensionality reduction techniques. These methods collect a number of interesting data fea tures, including covariance, input-output relationships, dynamical structure, and correlation between data sets. Mapping a set of high dimensional data features onto low dimensional data is the process of dimensionality reduction [10]

3. Semi-supervised learning: is often a combination of the first two approaches. The system trains on partially labeled input data—usually a lot of unlabeled data and a little bit of labeled data. Some application areas where semi-supervised learning is used include facial recognition, machine translation, fraud detection, labeling data, and text classification. [7]

4. Reinforcement learning: is a type of machine learning in which an agent learns to behave in an environment by trial and error. The agent receives rewards for taking actions that lead to desired outcomes and penalties for taking actions that lead to undesired outcomes. Over time, the agent learns to take actions that maximize its expected reward, is often used to train agents to solve complex problems in simulated environments, such as playing video games or controlling robots. However, reinforcement learning can also be used to train agents to solve problems in the real world, such as trading stocks or managing traffic. [7]

1.3 Computer vision:

Computer vision is a dynamic field dedicated to equipping machines with the ability to interpret and comprehend visual information from their surroundings. Images are fundamental to this process, serving as the primary source of visual data. A deep understanding of images and their characteristics is crucial for developing computer vision applications like object detection, facial recognition, and image segmentation. A computer vision image is essentially a digital representation of a visual scene, whether captured by a camera or created through software, with pixel values encoding the color and intensity of each point within that scene. [11]

1.4What is Image?

Images, whether captured by cameras or generated by computers, serve as two-dimensional representations of visual data. These computer vision images can be either grayscale, representing only light intensity, or color, encoding the combination of various colors within the visual scene. [11]

1.4.1 Magnetic Resonance Imaging (MRI)

MRI is generally the preferred modality for brain tumor diagnosis and treatment planning because of its high resolution and excellent soft tissue contrast [12] **(Magadza & Viriri, 2021)**. It can reveal the structure, metabolism, and function of tissues and organs in a non-invasive manner [13] **(Tjahyaningtijas, 2018)**. MRI is especially good at differentiating between different types of soft tissues in the brain, which helps in identifying the tumor, determining its size and location, and assessing its impact on surrounding structures [14] **(Elshaikh et al., 2021)**

1.4.2Computed Tomography (CT)

CT scans are also used to study brain tumors, also CT scans are often more readily available and cost-effective than MRI ,fortmore CT scans are particularly useful for visualizing the bones of the skull and detecting calcifications or hemorrhages within the tumor [14] **(Elshaikh et al., 2021)**

1.5 Brain tumor

A brain tumor is a mass of cells that have grown and multiplied uncontrollable i.e. a brain tumor is an uncontrolled growth of solid mass formed by undesired cells either normally found in the different part of the brain such as glial cells, neurons, lymphatic tissue, blood vessels, pituitary and pineal gland, skull, or spread from cancers mainly located in other organs.

Brain tumors are classified based on the type of tissue involved in the brain, the positioning of the tumor in the brain, whether it is benign tumor or malignant tumor and other different considerations. Brains tumors are the solid portion permeate the surrounding tissues or distort the surrounding structures. [4]

There is different type of brain tumor they are:

i) Gliomas, ii) Medulloblastoma, iii) Lymphoma, iv) Meningioma, v) Craniopharyngioma, vi) Pituitary adenoma.

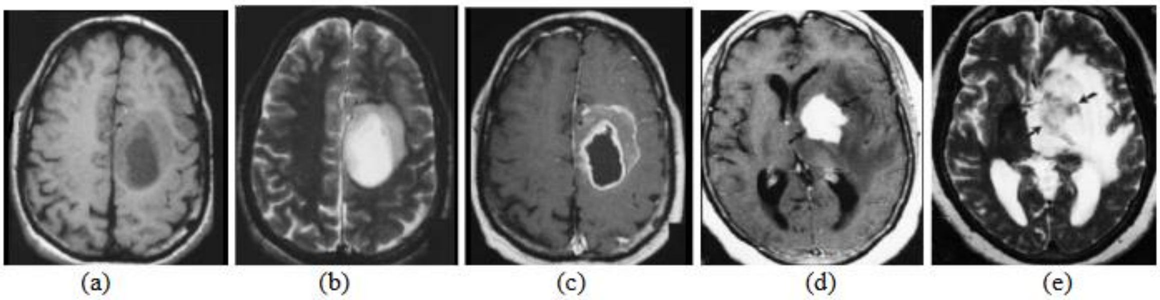


Figure 1:A set of brain tumor images from MRI of brain output cited by Herbert H. Engelhard et al.(2003)

- **a) Axial T1-weighted with tumor,**
- **b) T2-weighted with central positioning tumor,**
- **c) Contrast enhanced T1-weighted image showing ring formed tumor,**

d) **Contrast enhanced T1-weighted image with high grade oligodendro glioma**

e) **T2-weighted image with high grade oligodendro glioma from the same patient.**

1.5.1Automated system (CAD)

Automated system (detection) of brain tumor through MRI is basically called Computer-Aided Diagnosis (CAD) system.

The CAD system can provide highly accurate reconstruction of the original image i.e. the valuable outlook and accuracy of earlier brain tumor detection.

It consists of two or more stage. In the initial stage pre-processing has required after that stages post-processing i.e. segmentation are required. Then detection strategies and other information, feature extraction, feature selection, classification, and performance analysis are compared and studied.

Pre-processing techniques are used to improvement of image quality and remove small artefacts and noise for the accurate detection of the undesired regions in MRI.

Post-processing is used to segment with different strategy the brain tumor from the MRI of brain images. [4]

1.6 Deep Learning:

Deep learning is a subfield of machine learning that uses multi-layer artificial neural networks [15] **(Scanci, 2018)**. Deep learning is a machine learning technique that leverages multiple layers of nonlinear information processing to extract and transform features. Each subsequent layer uses the outputs of the previous layer as inputs [16] **(Vakalopoulou et al., 2023).**

In recent years, deep learning has been successfully applied to numerous problems across a wide range of application areas. These include natural language processing, sentiment analysis, cybersecurity, business, virtual assistants, visual recognition, healthcare, robotics, and many more [15] **(Scanci, 2018)**.

1.6.1 Deep Learning Architectures

Why Convolutional Neural Networks is more considerable over other classical neural networks in the context of computer vision?

Automatic Feature Extraction: CNNs can automatically learn relevant features from images, reducing the need for manual feature engineering. With the raw image as input, neurons in convolutional layers can detect edges, colors, and textures

Spatial Hierarchy Learning: Multiple convolutional layers can learn increasingly complex features. Lower layers might learn edges and corners, while higher layers combine these to recognize objects or parts of objects.

Translation Invariance: CNNs use convolutional filters that are shared across the entire image. This makes them translation-invariant, meaning they can recognize an object regardless of where it appears in the image.

Parameter Efficiency: CNNs typically have fewer parameters than fully connected networks, especially for large images. This is due to weight sharing and local connectivity, which reduces the computational cost and memory requirements.

1.6.2Convolution neural network (CNN):

A CNN consists of mainly three types of layers: convolutional layers, pooling layers, and fully connected layers**.** Training the network can be divided into forward and backward stages, In the forward

stage, the input image is classified depending on its weights and bias for each layer. The loss cost is calculated from the input data by using the predicted output. In the backward stage, depending on the loss cost measured, the gradients are calculated for each parameter
[17] **(Sinha et al., 2018)**

1.6.3 Neural network

1. Convolutional Layer:

Convolutional layers, the fundamental building blocks of CNNs, apply a series of learnable filters to the input image, with each filter designed to detect specific features. The output of a convolutional layer is a set of feature maps, which are then passed on to subsequent layers in the network, Main advantages of convolution layers are:

- **Sparse Connectivity**: A CNN has sparse connectivity, meaning that few weights are shared between two layers, whereas in a fully connected neural network, every neuron in one layer connects with every other neuron in the following layer. It is memory-efficient because it reduces the number of connections or weights, we require and the amount of memory needed to store those weights. Furthermore, the dot (.) operation is less expensive to compute than matrix multiplication.
- **Weight Sharing**: In CNN, all weights are applied to every pixel in the input matrix rather than creating specific weights between two neurons in adjacent layers. We can learn a single set of weights for all inputs rather than learning fresh weights for each neuron, which significantly reduces the training time as well as the other costs.
- Correlation between neighboring pixels is easy due to local connectivity.

What is a kernel?

A kernel is represented as a grid of discrete values or numbers, each of which is the kernel’s weight. All of a kernel’s weights are initially assigned random numbers when a CNN model is first being trained; alternative methods for initializing the weights are also available. After that, the weights are adjusted with each training epoch, and the kernel gains the ability to extract significant features [18]

2. Pooling Layer:

Pooling layers sub-sample feature maps, shrinking larger ones to lower ones while preserving dominant features. Different techniques include max pooling, min pooling, average pooling, gated pooling, and tree pooling. Max Pooling is the most popular technique. However, the main drawback of pooling layer is that it sometimes decreases the overall performance of CNN performance as it helps identify features in input images without considering their position[18].

Activation Functions (Non-Linearity):

Activation functions introduce non-linearity into neural network models by mapping input to output through the calculation of the weighted sum of neuron inputs plus a bias. In CNN architectures, these non-linear activation layers are strategically placed after learnable layers, enabling more complex learning and non-linear mapping capabilities, without non-linarites, the entire neural network would behave like a single linear layer, unable to learn intricate relationships in the data. Here are some common activation functions:

1.Sigmoid:

Real numbers are the input of the sigmoid activation function, which binds the output to the interval [0,1]. The sigmoid function has a "S" shaped curve The mathematical representation of sigmoid is:

$$f(x)sigm = 1/(1+e^{(}-x))$$

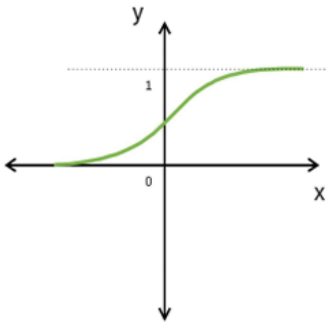


Figure 2: Sigmoid function

2.Tanh:

The input values (real numbers) are bound within the interval [-1, 1] by the Tanh activation function. Tanh is represented mathematically as follows:

$$f(x) = ((e^x - e^{(}-x)))/((e^x + e^{(}-x)))$$

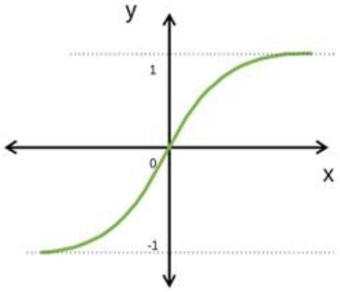


Figure 3:Tanh function

3.RELU:

Outputs the input directly if it is positive, otherwise, it outputs zero. ReLU is computationally efficient and has been shown to work well in many applications

$$f(x)ReLU = max(0,x)$$

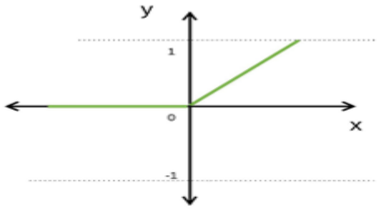


Figure 4:RELU function

3. Fully Connected (FC) Layer

Fully connected layers are typically placed at the end of a CNN. Each neuron in a fully connected layer is connected to all the activations in the previous layer [19] **(Srinivas et al., 2016).**

Also fully connected layers is the final layer of CNN consists of 90% of the parameters. The feed forward network forms a vector of a particular length to follow up processing. Since these layers contain most of the parameters, there is a high computational burden while training the data[20].

Loss Functions:

Now we have reached the output layer, which performs the final classification, is the last layer in any CNN architecture that is based on classification. Using a loss function, we compute the prediction error that the CNN model produced over the training set in this output layer. This prediction error informs the network of how far off its predictions are from the actual output, and it is then optimized as the CNN model learns. The estimate output of the CNN model, also referred to as the prediction, and the actual output, also referred to as the label, are the two parameters that the loss function uses to calculate the error. Different kinds of loss functions are applied to various kinds of problems. The following subsections provide a brief description of some of the most popular loss functions

1. Euclidean Loss Function (mean squared error)

In regression problems, the Euclidean loss, also known as mean squared error, is frequently utilized. is the mean squared error between the predicted output p and the actual output y, in each neuron of the output layer of CNN is defined as H(p,y)= (p– y)2. So, if there are N neurons in the output layer [18] then, the estimate Euclidean loss is defined as:

$$H(p, y) = 1/2N_(i = 1)^N (pi - yi)^2$$

Figure 5:Euclidean Loss Function

2Cross-Entropy or Soft-Max Loss Function

Cross-entropy loss, also called log loss function is widely used to measure the performance of CNN model, the probability p 0, 1 is its output. It is frequently used in multi-class classification problems as an alternative to the squared error loss function. It generates the output within a probability distribution by using SoftMax activations in the output layer. cross-entropy loss can be defined as:

$$H(p, y) = -_{i} y_{(} i) log(p_{i})$$

Figure 6:Soft-Max Loss Function

where i [1, N], p is the probability for each output category and y denotes the desired output and the probability of each output. [18]

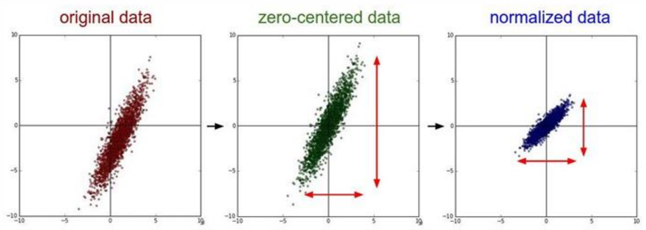
1.7 Training process of Convolutional Neural Network

the training process of Convolutional Neural Network involves in several steps include:

1.7.1 Data pre-processing

Mean-subtraction (Zero centering) and S-Normalization:

$$X' = X - x^{*} \qquad x^{*} = \frac{1}{N}\sum_{i=1}^{N} x_{i} \qquad X'' = \frac{X'}{\sqrt{\frac{\sum_{i=1}^{N}(x_{i}-x^{*})^{2}}{N-1}}}$$



The mean of the new values will be 0 and their standard deviation will be 1.

Figure 7:Mean-subtraction (Zero centering) and S-Normalization

1.7.2 the problem of overfitting and using of Regularization to CNN

Overfitting is a typical issue in machine learning in which a model performs well on training data but badly on unknown data. It occurs when the model is too complicated and has too many parameters in comparison to the quantity of training data, causing it to learn patterns particular to the training data rather than universal patterns that will generalize to unknown data.

Regularization

Regularization is a strategy for combating overfitting that involves adding a penalty to the objective function of the model to prevent it from learning patterns that are overly unique to the training data. The objective is to strike a compromise between a model that is complex enough to fit the training data effectively and one that is not so complicated that it begins to learn patterns that are particular to the training data and do not generalize to unobserved data.

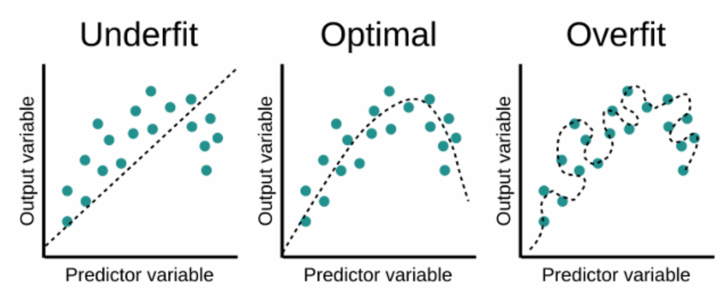
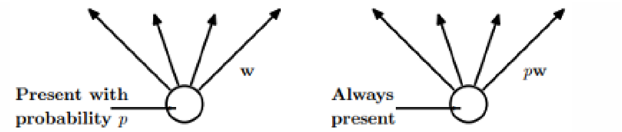
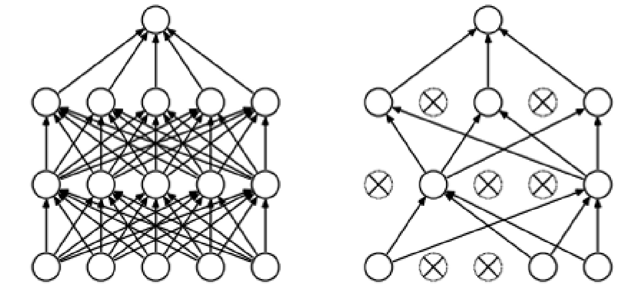


Figure 8:underfiting , optimal and overfitting cases

1. Dropout: One of the most popular regularization methods is dropout [21] during each training epoch, we randomly remove neurons from the network. By dropping the units (neurons) we try to distribute the feature selection power to all the neurons equally and we forced the model to learn several independent features. When a unit or neuron is dropped, it prevents it from participating in either forward or backward propagation during training. However, in the testing process, prediction is carried out using the full-scale network.



(a) At training time (b) At test time



(a) Standard Neural Net (b) After applying dropout.

Figure 9:Dropout

2. Drop-Weights: It is extremely comparable to dropout. The sole distinction is that in this case, we randomly remove the weights (the connections between neurons) during each training epoch rather than the neurons themselves.

3. L 2 Regularization: One of the most popular regularization techniques is the "weight decay" or 2 regularizations [22] Through the addition of a penalty term to the loss function equal to the "squared magnitude" of the coefficient, it compels the network's weights to decay towards zero (but not equal to zero). It regularizes the weights by heavily penalize the larger weight vectors. This is done by adding 1/2 ‖w‖ˆ 2 to the objective function, where is a hyper-parameter, which decides the strength of penalization and ‖w‖ denotes the matrix norm of network weights

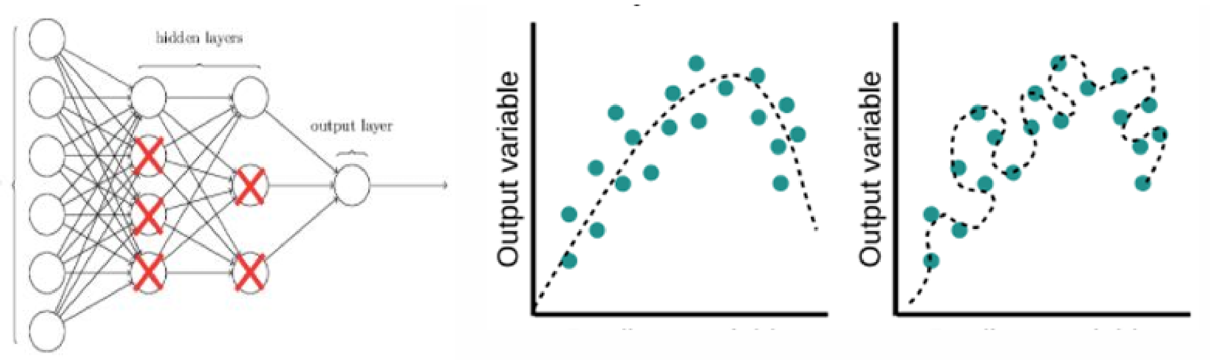


Figure 10: L 2 Regularization

4. L1 Regularization: The l1 regularization [22] is practically similar to the l2 regularization and is also commonly used in practice; the only difference is that the absolute value of the magnitude of the coefficients is used as a penalty to the loss function here, rather than the "squared magnitude" of the coefficients. Thus, the objective function with l1 regularization is as follows: Cost Function = loss + ‖w‖
5.

5. Data Augmentation: Training the model on a large amount of data with multiple variables is the simplest way to prevent over-fitting. This can be accomplished by employing data augmentation, in which the size of the training dataset is artificially increased using a variety of strategies such as: cropping, zooming...etc.

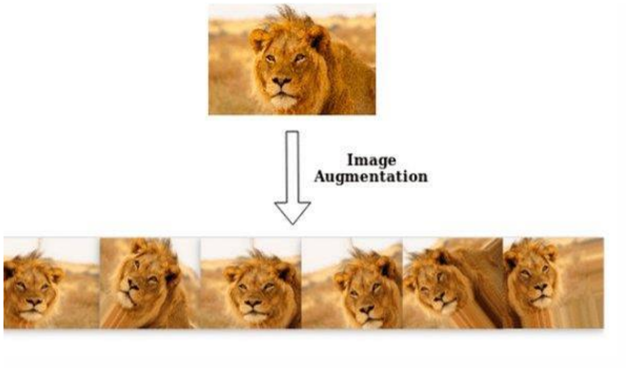


Figure 11: Data Augmentation

6. Early Stopping: In early stopping, we keep a small part (maybe 20% to 30%) of the train dataset as the validation set which is then used for Cross-Validation purposes, in which the trained model's performance is assessed across the validation set during each training epoch.

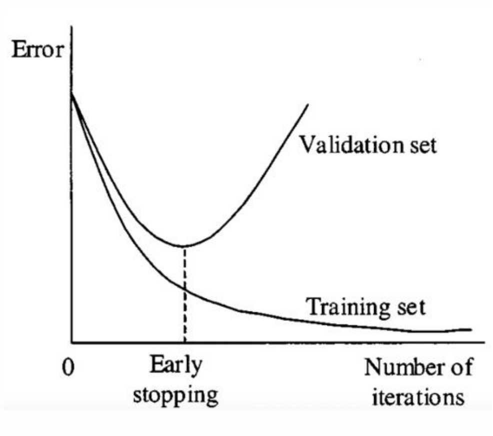


Figure 12: Early Stopping

7. Batch Normalization: Batch normalization [23] normalizes the output at each layer by subtracting the mean and dividing by the standard deviation, thereby explicitly ensuring that the output activations of a network will follow a unit Gaussian distribution. It also avoids the vanishing gradient problem.

1.8 Recent advancement in CNN Architectures

1.81 AlexNet:

motivated by Krizhevky et al.'s LeNet. created the first large-scale CNN model in 2012, known as AlexNet [24], with the purpose of classifying ImageNet data. Eight weighted (learnable) layers make up this structure; the first five are convolutional layers, and the final three are fully connected layers. The final output layer uses 1,000 units to classify the input images into one of the thousand classes of the ImageNet dataset because it was created for ImageNet data. Fig below displays the AlexNet architecture

1.8.2 ZFNet:

Zeiler and Fergus introduced ZFNet [25]at ECCV-2014, it has almost similar architecture as AlexNet except that here they used 7×7 filter with stride 2 in 1'st convolutional layer. When it comes to AlexNet, Krizhevky et al. use 11×11 filter with stride 4 filter in the first convolutional layer. As a result, ZFNet surpasses AlexNet in efficiency and wins the ILSVRC-2013

1.8.3 VGGNet:

Simonyan and Zisserman introduced one of the most popular CNN architectures in 2014, VGGNet [26] Six different CNN configurations were presented by the authors; the two most popular ones are VGGNet-16 (configuration D) and VGGNet-19 (configuration E). Figure illustrates the VGGNet-16 architecture

1.8.4 GoogLeNet:

Unlike all the previously discussed conventional CNN models, the GoogleNet [27] architecture employs network branches rather than single line sequential architecture. Szegedy et al. proposed the GoogleNet. in 2014. The GoogleNet uses the "Inception Module" as the funda mental building block of the network and has 22 weighted (learnable) layers. Each (simple basic) module consists of 1 × 1, 3 × 3, and 5 × 5 filtered convolution layers processed in parallel in the network. Their output feature maps are then combined to produce very high-dimensional feature output. They employed the inception module with dimensionality reduction to address this problem in place of the simple (basic) Inception module in their network architecture

1.8.5 ResNet:

He et al. notes that vanishing gradient issues plague a deep CNN model. from Microsoft proposed the ResNet model, which introduced the concept of "identity skip connection" to solve the vanishing gradient problem. Instead of learning a direct mapping (H(x) = F(x)), the ResNet architecture uses residual mapping (H(x) = F(x) + x). These bocks are referred to as residual bocks. The entire architecture of ResNet consists of multiple residual blocks with three-by-three convolution layers. Fig. The distinction between a residual mapping and a direct mapping is shown in figure

1.8.6 DenseNet:

DenseNet [29] illustrates how the concept of residual mapping is expanded by propagating each block's output to all succeeding blocks inside each dense block in the network. The information is propagated both forward and backward during the model's training process, strengthening the ability to propagate features and resolving the vanishing gradient issue. Huang and associates presented DenseNet. it emerges as the ILSVRC-2016 champion in 2016. Figure 2.21 displays a model that uses DenseNet.

1.9 Applications Areas of CNNs

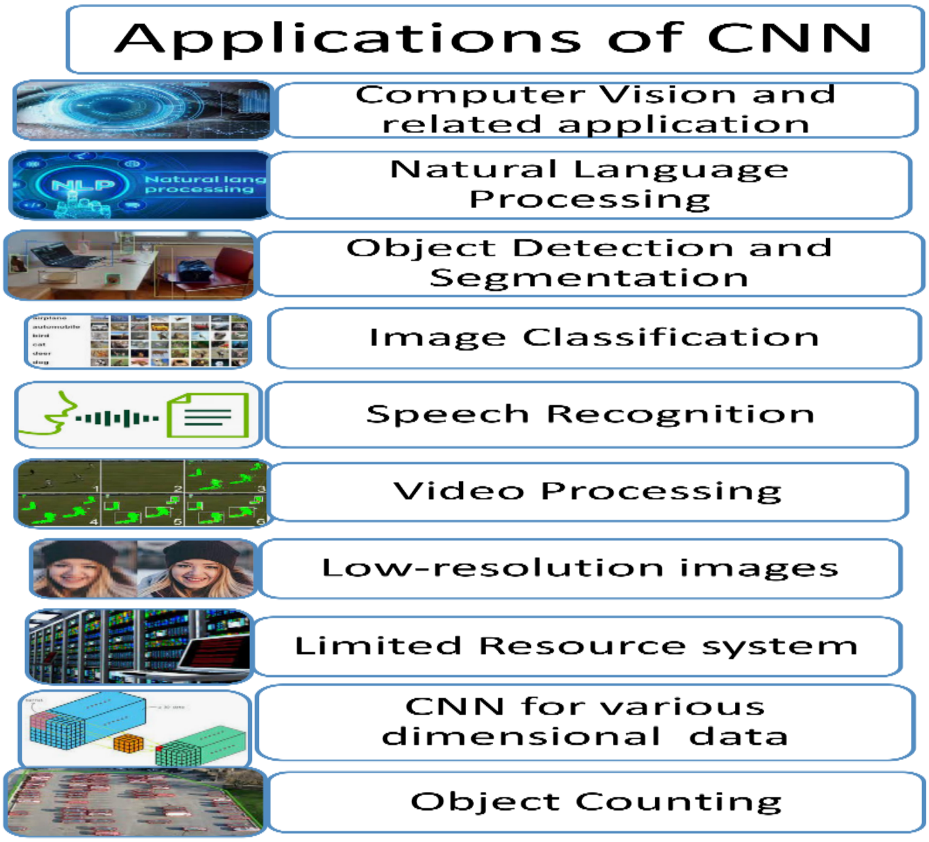


Figure 13: 1.9 Applications Areas of CNNs

1.10 Transfer learning

Basics of Transfer Learning: Transfer Learning utilizes information obtained from a primary task, usually on a vast dataset, to improve the effectiveness of a similar task with restricted data. The fundamental concept involves transferring acquired representations from the original task to the new task, taking advantage of the pre-trained model's ability to generalize. Important principles and methods in Transfer Learning comprise:

1. Pre-trained Model: Pre-trained Models are neural network models that have been trained on extensive datasets, like ImageNet, that consist of numerous labeled images. These models have the ability to extract valuable and general features from the input data. Some popular pre-trained models include VGG, ResNet and Inception. They are typically trained for tasks such as image classification, object detection, or language modeling. **[30]**

2. Feature Extraction: A common technique in Transfer Learning involves utilizing a pre-trained model to extract features. By removing the task-specific layers at the end of the pre-trained model, the output of the remaining layers can be used as features for the new task. These features can then be inputted into a new classifier or model that is trained specifically for the target task. This method is beneficial when the target task has restricted labeled data.**[30]**

3. Fine-tuning: Fine-tuning is another widely used method in Transfer Learning. Instead of treating the pre-trained model as a static feature extractor, this approach involves updating some or all of the model's layers during training on the target task. By adjusting the model's parameters through fine-tuning, we tailor the learned representations to better suit the unique characteristics of the target task. To prevent loss of pre-trained knowledge, fine-tuning is often conducted with a reduced learning rate.**[30]**

4. Freeze and Unfreeze: When adjusting a pre-trained model, it is typical to lock the initial layers (those near the input) and solely modify the subsequent layers. The initial layers capture basic features that are broadly applicable and transferable, whereas the later layers capture more specialized information for a specific task. Freezing the early layers helps to preserve the generic representations and prevents them from being overwritten during training. **[30]**

5. Domain Adaptation: Domain adaptation involves utilizing transfer learning in situations where the source and target tasks are related but have varying distributions. For example, a model initially trained on images of cats and dogs may struggle when applied to a different dataset containing wildlife images. To address this issue, techniques like domain adaptation and domain adversarial training can be employed to align the feature distributions 39 in both domains, facilitating more effective knowledge transfer.**[30]**

6. Selection of Transfer Layers: Choosing the right transfer layers depends on how closely related the source and target tasks are. Higher-level layers closer to the task-specific output tend to capture specialized features, while lower-level layers capture more general features. This decision is crucial and should be informed by a thorough understanding of the problem and available data **[30]** Transfer Learning is a popular technique in machine learning that utilizes pre-trained models to transfer knowledge between tasks. This allows us to take advantage of the labeled data and resources used in training these models, leading to strong performance even with small amounts of data. By grasping the fundamentals of Transfer Learning, individuals can successfully apply this method to various machine learning challenges

2.11 Related works

Machine Learning Approaches

While deep learning methods are increasingly favored, machine learning techniques have also demonstrated utility in brain tumor diagnosis. In this context, **Chatterjee et al. (2022)** [31] conducted a meta-analysis revealing that machine learning algorithms exhibited superior aggregate sensitivity (80%) and specificity (83.14%) compared to conventional MRI (81.84% and 74.78%, respectively). The meta-analysis highlighted high pooled sensitivity (0.926) and specificity (0.991) for ML algorithms. *This provides strong evidence for the effectiveness of machine learning, where the meta-analysis aggregates results from multiple studies, increasing the reliability of the findings. Also, the comparison to conventional MRI highlights that machine learning can offer improved diagnostic accuracy, demonstrating the overall potential of machine learning in this field.* Other studies have explored algorithms including artificial neural networks (ANNs), k-nearest neighbors (KNNs), decision trees (DTs), support vector machines (SVMs), and naive Bayes (NBs) **Zia et al. (2018)** [32] specifically found that ANNs provided the most accurate results, especially for medium to large datasets, demonstrating the effectiveness of ANNs as a machine learning technique. *This exploration expands the scope by showing that many different machine learning algorithms have been investigated, suggesting that there is no single "best" algorithm and that the choice depends on the specific application.* Moreover, **Farooqui (2024)** [33] emphasized the efficacy of diverse ML methods in medical imaging, particularly MRI. **Ghosh & Kole (2021)** [34] similarly compared algorithms like SVM, KNN, random forests, and gradient boosting, highlighting the variety of machine learning approaches used. In some approaches, deep learning techniques are used for feature extraction, which are then fed into traditional machine learning classifiers. For example, **Sarkar et al. (2023)** [44] demonstrated that feature extraction using CNNs like AlexNet, when combined with classifiers such as BayesNet and random forest, has proven effective, with accuracies ranging from 86.25% to 100%. However, despite these promising results, a common limitation across these studies is the reliance on high-quality datasets and the need for careful model selection and validation to ensure reliable clinical application **(Farooqui, 2024)** [33]. Therefore, the need to address these limitations and further enhance diagnostic accuracy motivates the exploration of more advanced deep learning techniques capable of automatically learning complex features and generalizing across diverse datasets.

Novel CNN Architectures and Optimization

One study **[35]** presented a three-stage approach that included preprocessing (including skull stripping and data augmentation), classification using a lightweight deep neural network, and segmentation using a noise-modified fast-cortical-association model (FL-MSCM), achieving a high classification accuracy of 99.58% and a Dice similarity coefficient of 95.7%. This study demonstrates a comprehensive approach to brain tumor diagnosis, combining preprocessing, classification, and segmentation. The high accuracy and Dice similarity coefficient indicate the effectiveness of this approach. The use of a "lightweight deep neural network" is also noteworthy, indicating a focus on efficiency. This highlights the potential for multi-stage approaches in improving diagnostic outcomes. Similarly, **Ray et al. (2020)** **[36]** presented Lu-Net, a less complex deep neural network model for segmenting and classifying early-stage brain tumors, achieving an overall accuracy of 98% on a high-resolution MRI dataset, outperforming Le-Net and VGG-16. The advantage over Le-Net and VGG-16, two well-established image classification architectures, lies in the fact that the proposed model offers significant improvements in terms of performance and complexity. This comparison demonstrates the benefits of developing specialized architectures tailored for specific tasks. In the context of improving the performance of convolutional neural networks (CNNs), a study using the 2020 BraTS dataset [37] found that strategic parameter adjustments—specifically, changing the final layer parameter to softmax and the optimizer parameter to AdaMax, followed by converting it to RMSProp—significantly improved accuracy to 99.74%. This study highlights the importance of tuning CNN parameters to achieve high accuracy. Sometimes, even simple parameter adjustments can lead to significant improvements, and it highlights the sensitivity of CNNs to these parameters. This emphasizes the need for careful optimization and experimentation. Finally, **Naseer et al. (2021)** [38] emphasized early diagnosis through a CNN trained on the BR35H dataset and evaluated on six others. Their CNN-based computer-aided design (CAD) system, enhanced with geometric data augmentation and statistical standardization, achieved an average accuracy of approximately 98.8% and a specificity of approximately 0.99, even achieving 100% accuracy on the BTS and BD-BT datasets, surpassing existing systems. The high accuracy and specificity, along with its superiority over existing systems, indicate that this computer-aided design system represents a significant advance. It highlights the importance of using techniques such as geometric data augmentation, statistical normalization, and data preprocessing to achieve promising results. In conclusion, these studies highlight the efforts to improve convolutional neural network (CNN) architectures and training methodologies to improve brain tumor diagnosis from medical images.

Deep Learning and Transfer Learning

Deep transfer learning has emerged as a powerful technique for brain tumor classification, leveraging pre-trained models to overcome the limitations of small datasets and improve diagnostic accuracy. **Srinivas et al. (2022)** [39] conducted a comparative analysis of VGG-16, ResNet-50, and Inception-v3, demonstrating the effectiveness of fine-tuning these models for MRI-based brain tumor prediction. Their results highlighted the VGG-16 model's ability to generalize well to unseen data after fine-tuning. Similarly, **Ahmad et al. (2022)** [40] explored the performance of seven pre-trained models, including VGG-16, VGG-19, ResNet50, InceptionResNetV2, InceptionV3, Xception, and DenseNet201, combined with traditional classifiers. They found that VGG-19 paired with SVM achieved a high accuracy of 99.39%, underscoring the potential of transfer learning even with limited data. **Badjie and Ülker (2022)** [41] introduced a deep learning model using the AlexNet architecture, achieving an accuracy of 99.62% in classifying brain tumors from MRI images, showcasing the automation potential for timely and accurate detection. **Asif et al. (2022)** [42] further contributed to this field by proposing an automated deep learning system using Xception, NasNet Large, DenseNet121, and InceptionResNetV2. Their study revealed that the Xception model achieved impressive accuracy rates, reaching 99.67% on a larger dataset, although they also cautioned about potential overfitting and limited generalization. Addressing the challenges of tumor complexity, **Asiri et al.** [43] presented an improved model combining CNNs with ResNet50 and U-Net architectures for detection, classification, and segmentation, achieving high IoU and DSC scores, demonstrating superior performance in accurately classifying and segmenting tumor regions. These studies collectively highlight the promise of deep transfer learning in enhancing the accuracy and efficiency of brain tumor diagnosis using medical imaging.

Conclusion

Based on the studies reviewed, we've developed a clear vision for our proposed model and its intended goals. On one hand, the success of various architectures like VGG-16, AlexNet, and Xception confirms that there's a significant field for improvement and innovation, making the exploration of novel architectures an important priority. This is precisely what we aim to achieve through our thesis, contributing a valuable addition to the field.

Furthermore, we recognize the importance of balancing accuracy and complexity, while achieving high accuracy is a primary goal, we also pay close attention to simplifying the model, reducing the number of parameters, training time, and required computational resources. In some cases, a simpler model that achieves comparable performance may be more desirable from a practical standpoint.

finally, these studies provide strong support for our thesis by highlighting the success of transfer learning, the importance of benchmarking, the potential of exploring different architectures, and the benefits of hybrid approaches. We aspire to benefits these findings to justify our research choices and to position our proposed model within the broader landscape of deep learning for brain tumor diagnosis.

Chapter 2

Methodology

Introduction

Convolution neural networks (CNN) have shown their powerful ability in image recognition and classification. The purpose of this chapter is to provide an overview on our proposed method then we will explain each component separately.

2-1 Proposed Methods:

The proposed method can be divided into two phases, it achieved the best accuracy rate. We experimented with different hyper-parameters, such as the number of epochs, learning rate, and batch size. Here is a more detailed explanation of each hyper-parameter:

- **Number of epochs**: An epoch represents one complete pass of the entire training dataset through the learning algorithm. The number of epochs is a hyper-parameter that specifies how many times the model will iterate over the entire training dataset during training. While training for more epochs can allow the model to learn more complex patterns in the data, it also increases the risk of overfitting, where the model becomes too specialized to the training data and performs poorly on unseen data. Monitoring the model's performance on a validation set is crucial to determine the optimal number of epochs[45]

Our choice: During the training, we used several different steps, starting with 15, 20, 25, and 30. We noticed that the program required a minimum of 17 steps, and a maximum of 28 steps, and average is about 25-26 to achieve the best results. Therefore, we set the number of steps at 25

- **Learning rate**: The learning rate is a crucial hyperparameter that determines the step size at which the model's weights are adjusted during training. It dictates how much the model responds to the estimated error each time the weights are updated. A learning rate that is too high can cause the training process to diverge, while a learning rate that is too low can result in slow convergence or getting stuck in local minima. [45]

Our choice: We chose a learning rate of 0.001 that changes after every 3 steps if the validation-loss value is stable or worse.

we use also Adams optimizer (par default) to select perfect value of LR

Early Stopping and reduce learning rate

We employed both early stopping and a learning rate reduction strategy during the training of our model to prevent overfitting, improve generalization performance, and optimize the training process.

Early Stopping: Early stopping is a regularization technique used to prevent overfitting, where the model starts to memorize the training data instead of learning the underlying patterns. We monitored the model's performance on a separate validation dataset during training. If the validation loss (or another chosen metric) stopped improving or started to degrade for a certain number of epochs (defined by a 'patience' parameter, we select was 4 epochs , because we split dataset 70 ,15, 15, training , validation , testing respectively), we halted the training process. [45].

Justification: By stopping training when the validation performance plateaus or declines, we prevent the model from continuing to learn noise or specific characteristics of the training data that do not generalize well to unseen data. This helps to improve the model's ability to perform accurately on new, real-world examples. The 'patience' parameter allows for minor fluctuations in the validation loss, preventing premature stopping due to random variations.

reduce learning rate: addition to early stopping, we implemented a learning rate reduction strategy. Specifically, we used **ReduceLROnPlateau**, which automatically reduces the learning rate when a chosen metric (e.g., validation loss) has stopped improving.

Justification: Initially, a higher learning rate allows the model to make rapid progress towards a good solution. However, as the model approaches the optimal solution, a high learning rate can cause it to overshoot and oscillate around the minimum. Reducing the learning rate allows the model to take smaller, more precise steps, converging more closely to the optimal solution and avoiding oscillations. **ReduceLROnPlateau** dynamically adjusts the learning rate based on the validation performance, ensuring that the learning rate is reduced only when necessary. This adaptive approach is more effective than using a fixed learning rate schedule.

Combined Effect: The combination of early stopping and learning rate reduction provides a robust approach to training our model. Early stopping prevents overfitting by halting training when validation performance degrades, while learning rate reduction helps the model to converge more effectively to the optimal solution. These techniques work together to improve the model's **generalization ability** and **overall performance**

• **Batch size**: refers to the number of data samples used in each iteration to compute the gradient and update the model's weights during training. It represents a trade-off: larger batch sizes can lead to more stable training and potentially faster convergence, but they require more memory. Smaller batch sizes, on the other hand, introduce more noise into the training process, which can help the model escape local optima but may also lead to slower convergence. By experimenting with different-hyper parameters, we were able to achieve a high accuracy rate with this method [45]

Our choice:

-We used 8 batches. we will have model slowly but more accrued, but if we increased the number of batches to 16, 32, and 64, respectively, the model will be more fast but less accrued.

2.1.1 The implementation:

The implementation was developed in Python (versions 3.12.7) using the **visual studio code IDE**. Several libraries were employed to provide specific functionalities. An overview of the languages and libraries utilized in this implementation is presented below:

- We preferred to use our own device, simply because it is better in terms of RAM (16GB) and the type of processor Ryzen 5 5600g.

- Another reason we not to use Colab is that it is slower than a computer, as it takes much longer to load the data than it takes for the computer to run the entire code.

Python:

The entire code is written in Python, a popular programming language for data analysis and machine learning. Python is a popular high-level interpreted programming language that is easy to learn and understand. Guido van Rossum was the creator, and it was originally published in 1991. Python prioritizes readability of code and makes indentation a crucial component of its syntax, which facilitates writing and comprehension**.**

Libraries:

NumPy: NumPy is a fundamental library for numerical computation in Python. This implementation uses NumPy for array creation and manipulation**[45]**

Scikit-learn: It is a machine learning library that provides various tools for model selection and evaluation. In the code, scikit-learn is used to split the dataset into train and test sets using the **train_test_split** function. **[45]**

TensorFlow: is an open-source software library for high performance numerical com potation. Its flexible architecture allows easy deployment of computation across a variety of platforms (CPUs, GPUs, TPUs), and from desktops to clusters of servers to mobile and edge devices**. [45]**

Matplotlib: is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible. **[45]**

Seaborn: is a library for making statistical graphics in Python. It builds on top of matplotlib and integrates closely with pandas data structures. Seaborn helps to explore and understand data**[45]**

Tqdm: A progress bar library. Use it to visually track the progress of loops and other long-running operations in your code. It makes it easy to see how far along a process is and estimate the remaining time. **[45]**

OS: Provides functions for interacting with the operating system. You can use it for tasks like creating directories, listing files, getting environment variables, and running system commands. It's essential for file system operations and system-level tasks.

Glob: Finds all the pathnames matching a specified pattern according to the rules of Unix shell-style wildcards. It's primarily used to search for files and directories based on patterns (e.g., txt to find all text files). **[45]**

Imbalanced-learn (imblearn): A library specifically designed for dealing with imbalanced datasets in machine learning. It provides various techniques for resampling the data, such as over-sampling the minority class or under-sampling the majority class, to improve model performance. **[45]**

Collections: Python's **collections** module provides high-performance container datatypes that offer alternatives to the standard **list**, **dict**, set, and **tuple**. Key classes include **Counter**, useful for tasks like frequency analysis; **default dict**, which simplifies the handling of missing keys in dictionaries; and **deque**, a double-ended queue that enables efficient insertion and deletion at both ends. **[45]**

2.1.2 Data processing:

Prior to being input into a model, images undergo several processing steps, including **resizing** and **normalization**, **oversampling**

Resize:

This operation adjusts the dimensions of the original image, either reducing or decreasing it, for various reasons such as decreasing computational load or conforming to the input size requirements of a specific model (pre trained CNNS models). We resized the images due to their inconsistent shapes, which led to unequal feature extraction across the dataset.
Resizing offers a good **balance** between processing speed and maintaining image quality compared to other methods. **[45]**

Normalization:

Here, we normalize the dimensions of the data samples (from the training, validation, and test datasets) using Z-score normalization. This involves subtracting the mean of each dimension from the data points and then dividing by the standard deviation of that dimension. This process ensures that the data is centered around zero and scaled to have a unit standard deviation. **[45]**

Data Augmentation

We incorporated data augmentation techniques on 10% of our training dataset to improve the generalization ability and robustness of our binary classification model. Data augmentation involves applying various transformations to existing training samples to create new, synthetic samples, effectively increasing the size and diversity of the training data.

The decision to augment only 10% of the dataset was a deliberate choice to balance the benefits of increased data diversity with the potential risks of introducing artificial or unrealistic data patterns. Augmenting the entire dataset could potentially lead to overfitting on the specific data augmentation techniques used, rather than learning the underlying patterns in the original data. By applying data augmentation to a smaller subset, we aim to introduce sufficient variability to improve generalization without overwhelming the model with synthetic data.

The specific data augmentation techniques applied (e.g., rotations, flips, zooms, translations, noise addition)

Our choice:

Here we explain that we used vertical and vertical flipping?
Rotational flipping cannot be used because most of the data provided does not contain images with rotation.
Zoom in and out cannot be used because it negatively impacts the data, as the white space in the image can either be enlarged, making it difficult to distinguish between the infected and the healthy images, or reduced, making it difficult to identify the healthy images.
Moving can cause some parts to fall out of view, resulting in data loss.
Noise cannot be used because it may cause conflict between some healthy images and the infected images.
- By augmenting 10% of the dataset, we expect to achieve the following benefits:

Improved Generalization: The increased diversity of the training data will help the model learn more robust features and generalize better to unseen data.

Reduced Overfitting: Data augmentation acts as a regularizes, preventing the model from memorizing the training data and improving its ability to generalize to new examples.

Increased Robustness: The model will become more resilient to variations in the input data, such as slight rotations, translations, or noise, making it more reliable in real-world applications.

In summary, we chose to augment 10% of our dataset to strike a balance between increasing data diversity and avoiding overfitting, ultimately aiming to improve the generalization ability and robustness of our binary classification model

Oversampling:

Oversampling is a prevalent technique in data analysis and machine learning, particularly useful for handling imbalanced datasets. Imbalanced datasets arise when one class contains considerably fewer instances compared to another, potentially leading to biased models that perform well on the majority class but poorly on the minority class. ***we choice*** the next.

Random Oversampling (ROS):

Random Oversampling involves duplicating instances from the minority class at random to increase its representation within the dataset.

2.1.3 The activation function

we choice **ReLU (Rectified Linear Unit)** as the activation function for all hidden layers in my binary classification neural network, while employing a **sigmoid** activation function in the output layer. This combination was selected to leverage the strengths of each function in their respective roles

ReLU:

Firstly, ReLU's simple computational form (f(x) = max (0, x)) leads to faster training times compared to more complex activation functions like sigmoid or tanh. This is because ReLU avoids the computationally expensive exponential operations involved in those functions.

Secondly, ReLU helps to alleviate the vanishing gradient problem, which can hinder learning in deep neural networks. When the input to a ReLU neuron is positive, the gradient is 1, allowing gradients to flow more easily through the network during backpropagation. This is in contrast to sigmoid and tanh, where the gradients can become very small, especially for large input values, effectively stopping learning in earlier layers. **[45]**

sigmoid function:

In binary classification, the most common activation function used in the output layer is the sigmoid function

Here's why choice it?

Output Range: The sigmoid function outputs a value between 0 and 1. This is ideal for representing probabilities, where a value close to 1 indicates a high probability of belonging to the positive class, and a value close to 0 indicates a high probability of belonging to the negative class.

Interpretation: The output of the sigmoid function can be directly interpreted as the probability of the input belonging to the positive class.

Differentiability: The sigmoid function is differentiable, which is essential for training neural networks using gradient-based optimization algorithms.

While other activation functions exist, the sigmoid function is the standard choice for the output layer in binary classification tasks due to its probabilistic interpretation and suitability for gradient-based learning.

2.1.4 The loss and the optimizer functions that used

Across all models, we utilized the Binary Cross-entropy loss function, optimized and minimized via the Adam optimizer.

As previously mentioned, we will now explore two complementary concepts crucial to our model training. The first pertains to quantifying the model's error through a loss function, and the second focuses on enhancing and optimizing the model. These are the **Binary Cross-entropy loss function** and **the Adam optimizer.**

Binary Cross-entropy loss function

Binary cross-entropy, also known as log loss, quantifies the difference between the predicted probability distribution and the actual distribution of the target variable. In binary classification, the target variable can take on one of two values (0 or 1), representing the two classes. The model outputs a probability score between 0 and 1

The binary cross-entropy loss function penalizes incorrect predictions more heavily than correct predictions, with the penalty increasing as the predicted probability deviates further from the true label. Furthermore, binary cross-entropy is differentiable, which is essential for training neural networks using gradient-based optimization algorithms. The derivative of the loss function provides a clear signal for updating the model's parameters in the direction that minimizes the loss.

In summary, we chose binary cross-entropy because it is a well-established and effective loss function for binary classification tasks, providing a direct measure of the model's performance in predicting probabilities and guiding the optimization process towards accurate and reliable predictions [45]

Adam optimizer

The Adaptive Moment Estimation (Adam), is a popular optimization algorithm utilized in deep learning to adjust a model’s parameters throughout the training phase. This method involves adapting the learning rate and momentum terms for optimal performance. Adam combines the advantages of AdaGrad, which adjusts learning rates per parameter, and RMSProp, which utilizes a moving average of squared gradients [18]

a. Learning rates: Adaptive learning rates involve adjusting the learning rate based on gradient behavior during training to determine the optimal step size for parameter updates. This helps facilitate effective convergence and avoids getting stuck in local optima by finding the appropriate learning rate for each parameter.

b. Momentum: Momentum is a concept that signifies the acceleration of the optimization process by incorporating a moving average of past gradients. It gathers the gradients from previous iterations and incorporates a portion of the amassed gradient into the current gradient to update the parameters. The momentum term plays a significant role in deter mining the impact of previous gradients on the current update. In cases where gradients are noisy or the loss landscape is intricate, incorporating momentum can help optimization methods progress more smoothly towards the optimal solution and expedite the convergence process **[18]**

2.1.5 Deep learning using CNN architectures

This section focuses on the use of Convolutional Neural Network (CNN) architectures in deep learning. These networks are specifically designed to process data with a grid-like structure, such as images, and are characterized by their ability to automatically extract important features from the data. This is achieved through the use of convolutional layers and pooling layers, which allow the network to recognize spatial patterns in the data.

Function Variables

The function used to build the CNN includes a number of important variables that define the behavior of the network.

First, we have "**filter**," which represents the number of filters in each layer. This filter is multiplied by a constant in each layer, with the default being 1 filter.

Second, we have "**Size_fil**," which defines the size of each filter, with a default value of 3.

Third, we have "**Size_pol**," which defines the size of each maxpolin, with a default value of 2.

Finally, we have "**Learning rate**," which determines the learning rate, with a default value of None (adams).

Function Components

The "**simple_cnn**" function consists of several essential components that work together to form the CNN.

The function starts with a convolutional layer containing a "filter" multiplied by different values. This is followed by a ReLU activation function to introduce non-linearity, and then a max pooling layer to reduce the spatial dimensions and retain the most prominent features.

These two layers are repeated three times, allowing the network to learn complex features from the data.

Next, a flattening of the cells is followed by a dropout layer to prevent overfitting.

Finally, an output with a single cell and a sigmoid function as an activation function is produced.

2.1.6 Using transfer learning

Transference learning is a machine learning approach that takes advantage of pre-trained models to solve new tasks by transferring the knowledge. Transference learning utilizes pre-trained models to address novel tasks by transferring the expertise of the pre-trained model gained from past experiences and applying its weights to new models. These weights serve as a starting point for a new model, with additional layers being incorporated and their parameters updated to suit the new task.

Our choice:

We employed transfer learning, utilizing the pre-trained VGG16 and ResNet-50 convolutional neural network models, to leverage knowledge gained from large-scale image datasets for our brain tumor classification task (tumor/no tumor) using a dataset containing MRI and CT images. The selection of VGG16 and ResNet-50 was based on their proven performance, architectural characteristics, and widespread use in the computer vision community.

VGG16: is a classic convolutional neural network known for its simple and uniform architecture, consisting of multiple convolutional layers with small 3x3 filters, followed by max-pooling layers. Its strengths lie in its ability to learn intricate spatial hierarchies from image data.

- **We choice** VGG16 because its well-defined and relatively deep architecture has been shown to be effective in extracting relevant features from images. While more recent architectures exist, VGG16 provides a good balance between performance and computational cost. Its pre-trained weights, learned on ImageNet, offer a strong starting point for feature extraction, which can be fine-tuned for our specific brain tumor classification task. The consistent use of small convolutional filters allows it to capture fine-grained details in the MRI and CT images, which may be crucial for differentiating between tumor and non-tumor cases.

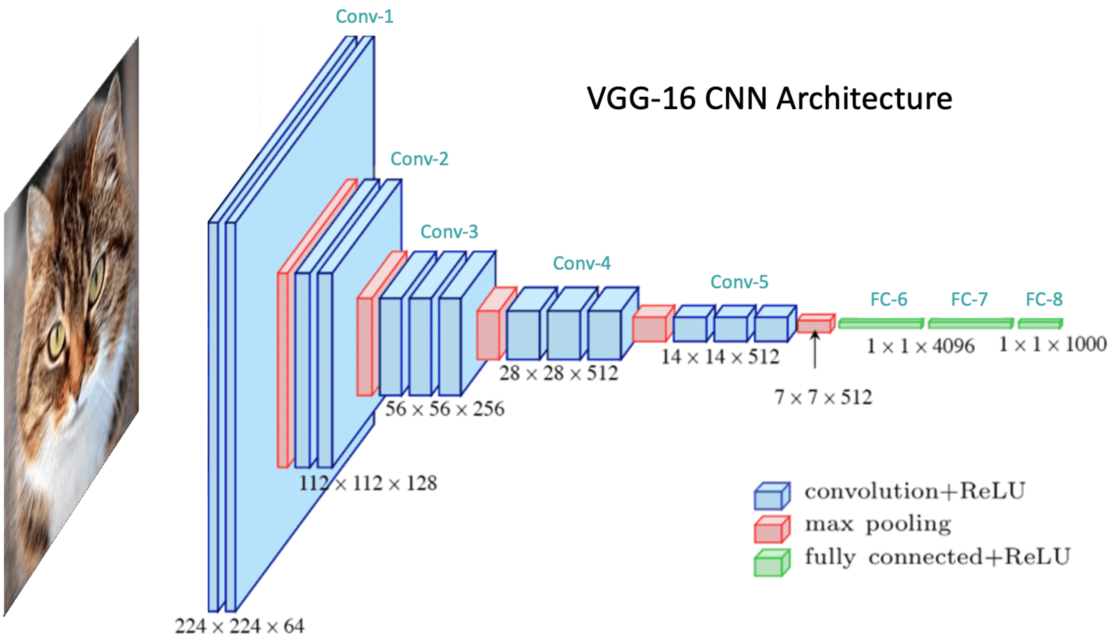


Figure 14: VGG 16 Architecture

ResNet-50: ResNet-50 (Residual Network with 50 layers) represents a significant advancement in deep learning architectures, addressing the vanishing gradient problem that can occur in very deep networks. ResNet-50 utilizes residual connections (skip connections) that allow the network to learn identity mappings, enabling the training of much deeper models.

- **We selected** ResNet-50 because its deep architecture and residual connections enable it to learn more complex and abstract features compared to shallower networks. The skip connections facilitate the flow of information through the network, allowing it to effectively train on more challenging datasets. The pre-trained ResNet-50 model has demonstrated excellent performance on various image classification tasks, and we hypothesized that its ability to learn hierarchical representations would be beneficial for identifying subtle differences between tumor and non-tumor regions in our MRI and CT images. Furthermore, its robustness to the vanishing gradient problem makes it a suitable choice for transfer learning, as it can be fine-tuned without significant degradation in performance.

Complementary Strengths: VGG16 and ResNet-50 offer complementary strengths. VGG16's simplicity and focus on spatial hierarchies provide a solid baseline for feature extraction, while ResNet-50's depth and residual connections enable it to learn more complex and abstract features. By comparing the performance of these two models, we can gain insights into the optimal architectural characteristics for our brain tumor classification task.

Availability of Pre-Trained Weights: Both VGG16 and ResNet-50 have readily available pre-trained weights on large datasets like ImageNet. This significantly reduces the training time and computational resources required to achieve good performance, as the models have already learned general image features.

In summary, the selection of VGG16 and ResNet-50 was driven by their proven performance, distinct architectural characteristics, and the availability of pre-trained weights, making them well-suited for transfer learning in our brain tumor classification task.

Chapter 3

Experimental Setup

Introduction

In this chapter, we evaluate how well performance of the proposed model We start by explaining the datasets we used, and the materials we used for our experiments. Then, we present the results for our contribution and discuss how well they performed. We consider evaluation metrics like accuracy, precision, recall, and how fast the algorithm runs. By doing this evaluation, we show how effective the proposed model is classifying brain tumor.

This chapter is important because it gives valuable information about our approach and its success.

3.1 Dataset Description

The dataset, created by **Murtozalikhon**, is a collection of multimodal medical images consisting of both Computed Tomography (CT) and Magnetic Resonance Imaging (MRI) scans of the brain.

It's designed to aid researchers and healthcare professionals in developing AI models for the automatic detection, classification, and segmentation of brain tumors. [46], The goal is to improve the efficiency and accuracy of brain tumor diagnosis, potentially leading to earlier and more effective treatment

The dataset includes high-resolution CT and MRI images captured from multiple patients, with each image labeled with the corresponding case (healthy, tumor)

CT: provides clear visualization of bone structures.

MRI: excels in visualizing soft tissue details, which is crucial for identifying and delineating tumors. such that Different MRI modalities like T1-weighted (T1), contrast-enhanced T1-weighted (T1c), T2-weighted (T2), and Fluid Attenuated Inversion Recovery (FLAIR) can reveal different aspects of brain tissue with varying degrees of clarity [46]

These factors (availability, multimodality, labeled data and Relevance), helped us to select this dataset. Availability: Publicly available on the Kaggle platform. [46]

Multimodality: Combining CT scans and MRIs.

Labeled data: Labeled images enable supervised learning (Classification task in our case)

Importance: This dataset directly addresses the problem of classifying brain images into healthy and diseased images.

- However, we found some limitations, such as:

Class imbalance: There are more infected examples than healthy examples in MRI images.

In this case, we used techniques such as **Oversampling** to avoid this problem.

Multiple resizing: where we standardized the size of all images by Increase or decrease their size, this step can reduce the quality of some images, or cause the loss of some important small details.

3.2 Performance Evaluation Metrics

3.2.1 Accuracy

is a fundamental metric for evaluating the performance of a classification model, providing a quick snapshot of how well the model is performing in terms of correct predictions. It is calculated as the ratio of correct predictions to the total number of input samples. [47]

$$\frac{TP + TN}{TP + FP + FN + TN}$$

Figure 15: Accuracy

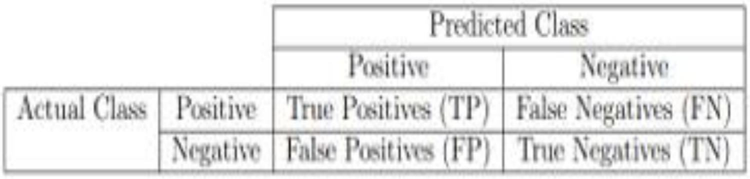
3.2.2 confusion matrix

The confusion matrix is one of the most common metrics, which is a square matrix that is used to evaluate the performance of a classification model and summarizes its performance by counting the number of correct and incorrect predictions for each class. [46]

The confusion matrix's components are described in the following order:

True Positives (TP): The number of situations where a positive outcome was accurately predicted. **False Positives (FP)**: The number of instances that are falsely identified as positive. True Negatives **(TN)**: The number of instances that are correctly predicted as negative. False Negatives

(FN): The number of situations that were mistakenly projected as negative



		Predicted Class	
		Positive	Negative
Actual Class	Positive	True Positives (TP)	False Negatives (FN)
	Negative	False Positives (FP)	True Negatives (TN)

Figure 16: confusion matrix

3.2.3 Sensitivity/Recall:

Recall is the ratio of correctly predicted positive instances to the total actual positive instances. It measures how well the model captures all relevant positive cases.

$$\frac{TP}{TP + FN}$$

Figure 17: Sensitivity/Recall

3.2.4 Precision:

is a measure of a model’s performance that tells as how many of the positive predictions made by the model are actually correct.

$$\frac{TP}{TP + FP}$$

Figure 18: Precision

3.2.5 F1 Score:

F1 Score is the harmonic mean of precision and recall, providing a fair assessment of both measurements. It is advantageous when classes are unbalanced since it combines precision and recalls into a single value. Calculated as :

$$\frac{2 \cdot (\text{Precision} \cdot \text{Recall})}{\text{Precision} + \text{Recall}}$$

Figure 19: F1 Score

3.2.6 Area Under Curve (AUC):

It is one of the widely used metrics and basically used for binary classification. The AUC of a classifier is defined as the probability of a classifier will rank a randomly chosen positive example higher than a negative example. Before going into AUC more, let me make you comfortable with a few basic terms.

3.3 Hardware and Software Configurations

Hardware

The system used for model training and experimentation is equipped with an AMD Ryzen 5 5600G processor (6 cores, 12 threads, base clock 3.9 GHz, boost up to 4.4 GHz), 16GB of DDR4 RAM running at 3000MHz, and a Kingston solid-state drive (SSD) for storage

3.4 Experimental results

Introduction: In this part, we show how different setups affected the model's performance. We tested **eight different ways of preparing the images**. Some used **color (RGB)**, some used **grayscale**, and others included **normalization**, **data augmentation**, and **resizing**. All images were **MRI brain scans** resized to **256×256 pixels**. The results are shown in the **three figures below**. At the end, we will also mention the performance of the model on **CT images**, focusing on accuracy.

3.4.1 Impact of Image Processing Settings on Model Performance (MRI):

This figure presents the model's performance using three metrics: **test accuracy**, **test loss**, and **training time**. These help us understand how well the model works on unseen data, how confident it is in its predictions, and how long it takes to train.

The Methods (in order):

1-grayscale without normalization
2- grayscale with normalization
3- grayscale with data augmentation and resizing without normalization
4- grayscale with data augmentation and resizing with normalization
5- RGB with normalization
6- RGB without normalization
7- RGB with data augmentation and resizing with normalization
8- RGB with data augmentation and resizing without normalization.

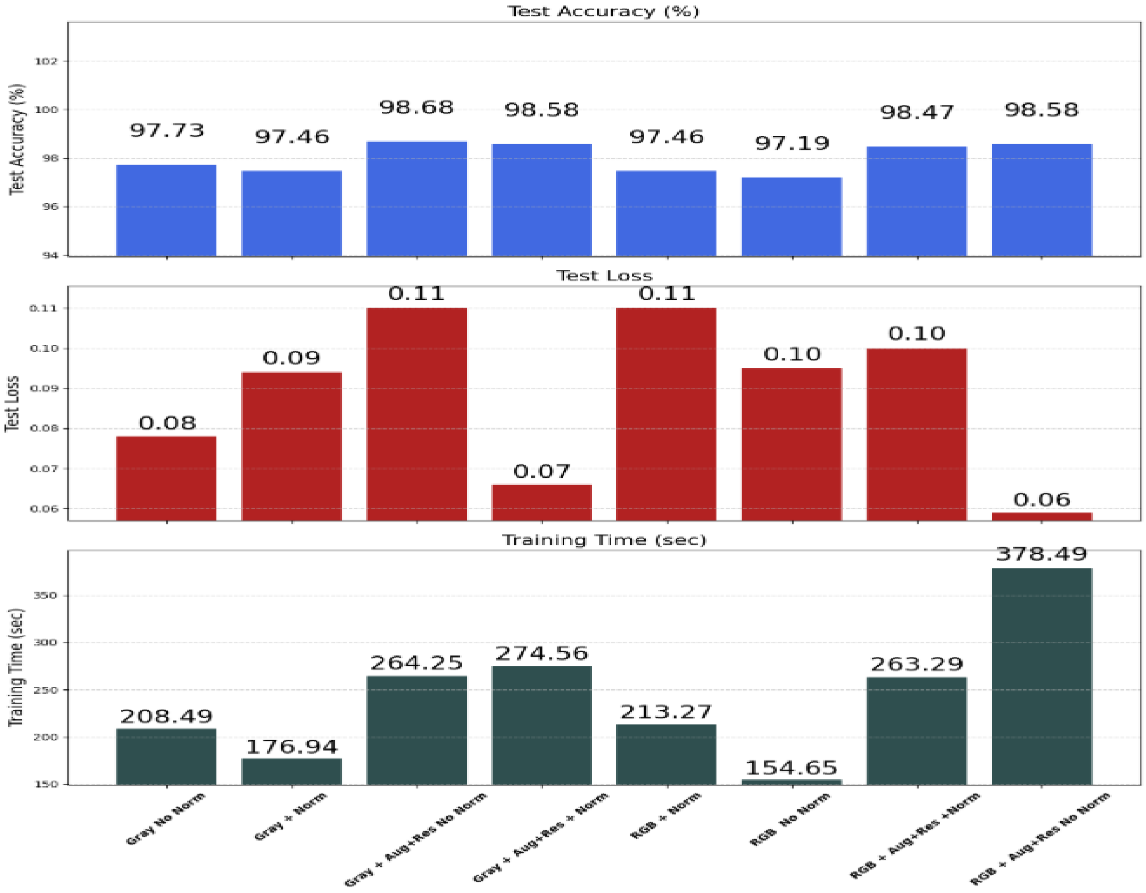


Figure 20: Test Accuracy, Test Loss, and Training Time

The best **test accuracy** is seen in the **RGB + Aug + Res + No Norm** setup, reaching **98.58%**, followed closely by the **Gray + Aug + Res + Norm** setup at **98.68%**. The **lowest test loss** is also found in RGB + Aug + Res + No Norm (**0.06**), meaning this model made fewer mistakes. RGB models took the most **training time**, especially with data augmentation and resizing. The **fastest training** setup was **RGB No Norm** at **154.65 seconds**, while the **slowest** was **RGB + Aug + Res + No Norm** at **378.49 seconds**. These results suggest that **RGB + Aug + Res + No Norm** gives the **best test performance**, although grayscale models are faster to train.

This figure compares three important scores: **precision**, **recall**, and **F1 score**. These help us measure how accurately the model finds the correct class, especially in tasks where detecting the right condition (like a tumor) is very important

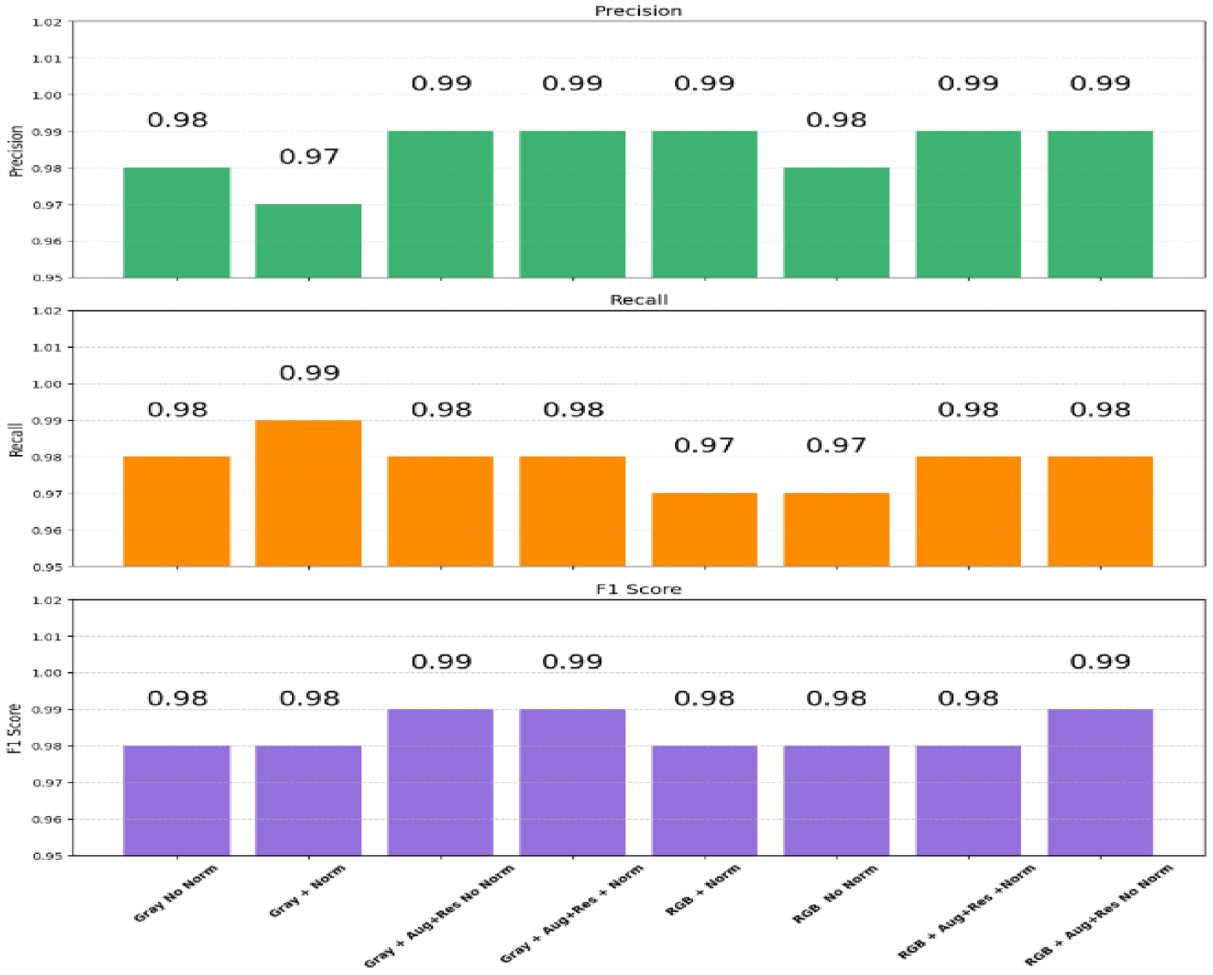


Figure 21: Precision, Recall, and F1 Score Comparison

.Precision is highest (**0.99**) in **Gray + Aug + Res + Norm** and **RGB + Aug + Res + No Norm**, showing that the model made fewer false positives. Recall is also highest (**0.99**) in **Gray + Norm** and **Gray + Aug + Res + No Norm**, meaning the model correctly found almost all the positive cases. The **F1 score**, which balances precision and recall, is best (**0.99**) in **Gray + Aug + Res + Norm** and **RGB + Aug + Res + No Norm**. These results show that these two setups provide the most balanced and strong performance.

This figure shows the model's **cross-validated accuracy**, **standard deviation**, and **total training time**. Cross-validation checks how stable the model is across different splits of the training data. A lower standard deviation means the model gives more reliable results.

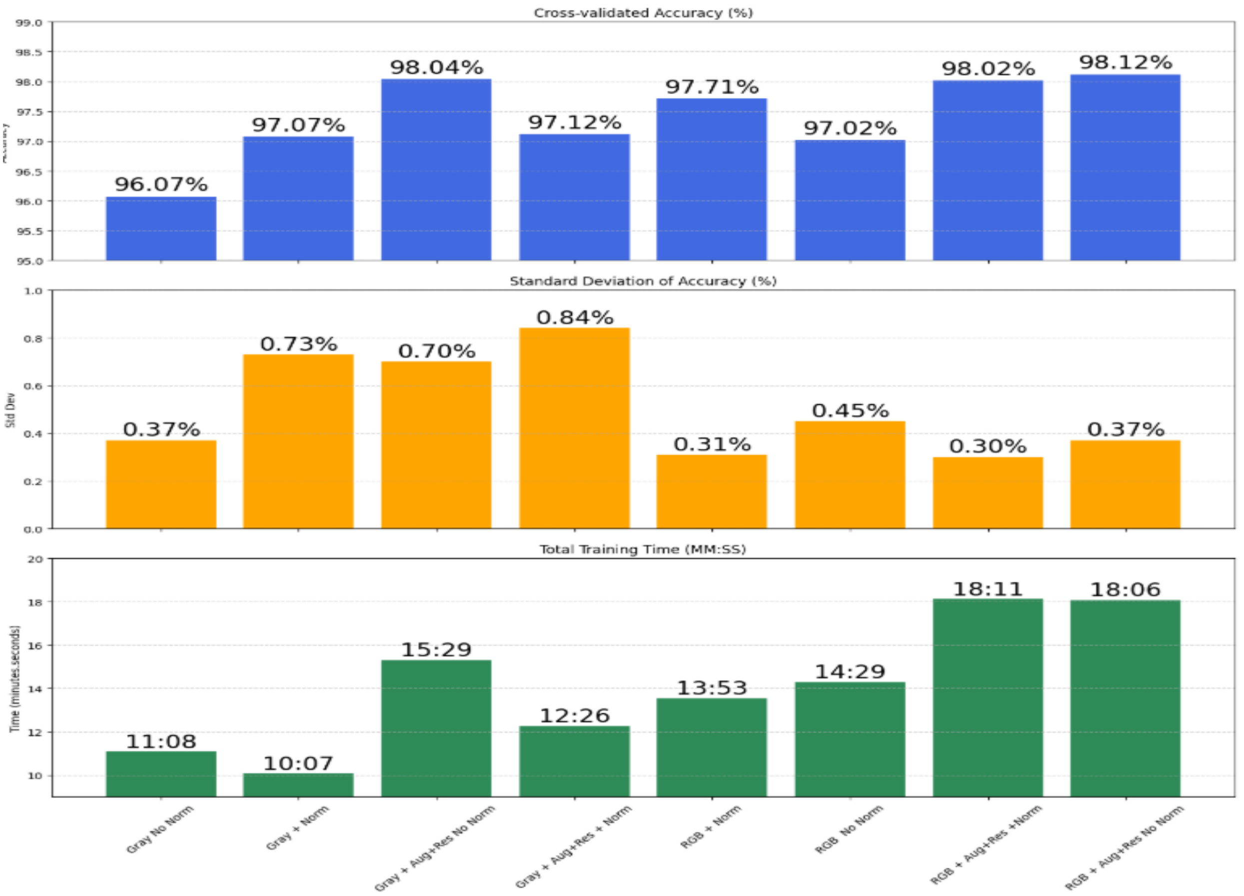


Figure 22: Cross-Validated Accuracy, Standard Deviation, and Training Time

The **highest cross-validated accuracy** is seen in **RGB + Aug + Res + No Norm** (**98.12%**), while **Gray + Aug + Res + Norm** also performed well (**98.04%**). The most **stable model**, with the **lowest standard deviation** (**0.30%**), is **RGB + Aug + Res + Norm**, meaning it gives consistent results across different runs. Training time is longest in RGB models, especially those with data augmentation. **Gray + Norm** is the fastest setup, needing only **10:07 minutes**. These results show that **RGB + Aug + Res + Norm** is the **most stable**, and **RGB + Aug + Res + No Norm** gives the **highest accuracy**, though they both take longer to train.

3.4.2 Model Performance on CT Images:

While all the figures above are based on MRI images, we also evaluated the model on CT images. The setup using data augmentation and resemblance without normalization achieved an accuracy of 99.15% with a loss of 0.043, and precision, recall, and F1 score all reached 99% in just 3 minutes and 21 seconds, demonstrating the model's strong ability to generalize across different types of medical images.

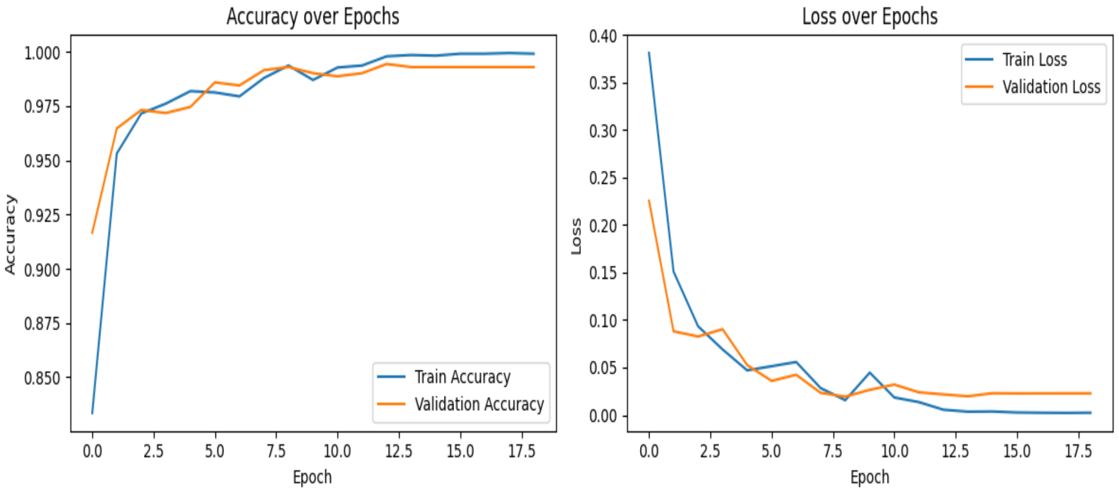


Figure 23: Accuracy and Loss over epoch of CT images

The model achieved a cross-validated accuracy of 98.88%, with a standard deviation of 0.33%, indicating consistent performance across folds. The mean AUC was 0.99 with virtually no variance (±0.00), reflecting excellent discrimination capability. The total training time 12 minutes and 53 seconds.

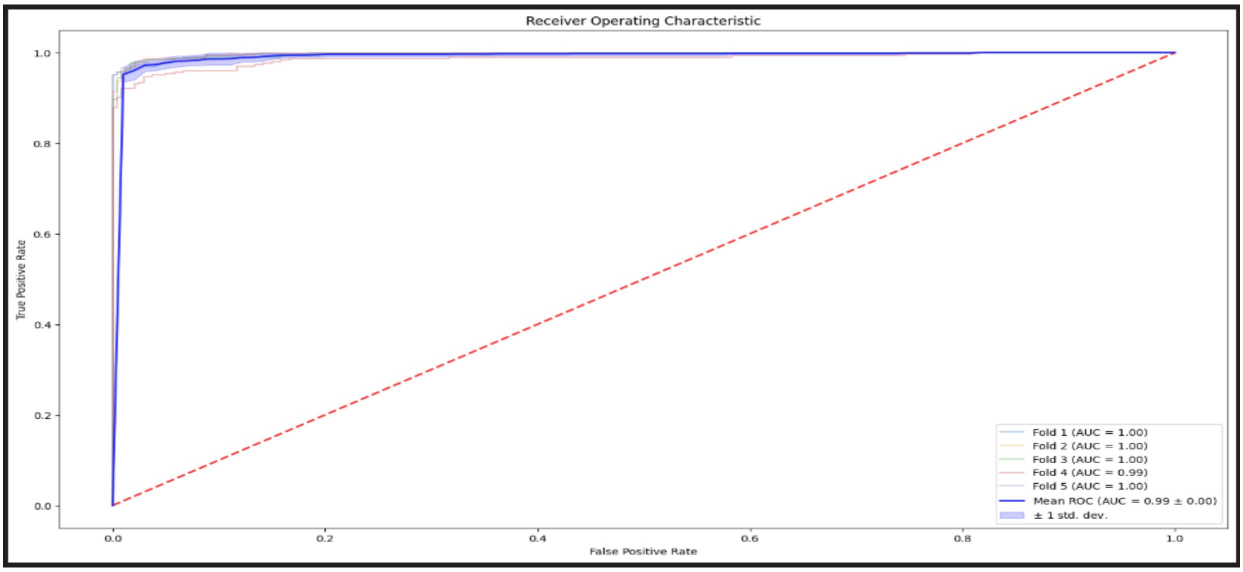


Figure 24: ROC Curve Cross-validation

3.4.3 Comparing Deep Learning Models for Brain Tumor Detection

In this part of our study, we looked at how different deep learning models perform when identifying brain tumors from MRI images. All the images were resized to 256×256 pixels and preprocessed using normalization, without any data augmentation. The training was done using VS Code on a regular computer, without GPU support.

We compared four models: VGG16, ResNet50, an existing Brain Tumor Classification model, and our own Simple CNN.

Model	VGG16	ResNet50	Brain Tumor Classification	Simple CNN (Our Model)
Accuracy	98.26%	98.53%	98.43 %	98.40 %
Precision	0.98 / 0.99	0.98 / 0.99	0.99/0.98	0.98/0.98
Recall	0.98 / 0.98	0.99 / 0.98	0.97/0.99	0.98/0.99
F1 Score	0.98 / 0.99	0.98 / 0.99	0.98/0.99	0.98/0.99
Test Loss	0.06	0.12	0.074	0.1
Training Time (min)	68:36	36:37	62:38	4:19
Total Params	16.81M	40.37M	228.2K	49.9K
Trainable Params	2.10M	16.78M	227.8K	49.9K
Non-trainable Params	14.71M	23.59M	480	0
Model Size	64.13 MB	153.98 MB	891.57 KB	194.82 KB

Tableau 1: Tableau 1: Table of comparions

ResNet50 showed the highest accuracy overall, meaning it was the best at making correct predictions. VGG16 also performed well, especially with the lowest test loss, which means it was quite reliable on new data. Interestingly, our Simple CNN—despite being much smaller and faster—also gave competitive results with high precision, recall, and F1 scores.

What stands out most about our Simple CNN is how efficient it is. It trained in under five minutes, while the others took much longer. It also has a very small model size—less than 200 KB—making it ideal for devices with limited resources.

In summary, if you're looking for the best possible accuracy and have the hardware to support it, ResNet50 or Vgg16 is a strong option. But if you need something fast, lightweight, and still highly accurate, our Simple CNN presents itself as a practical and balanced choice.

3.4.4 Discussions

Our research has shown that a lightweight CNN architecture can achieve high accuracy in classifying brain tumors from MRI images, while significantly reducing the computational cost compared to traditional deep learning models.

Interpretation of Results:

The improvement in efficiency is due to the optimized architecture of our CNN, in terms of reducing the number of parameters and computational operations required to process images. Additionally, the use of techniques such as data augmentation enhanced the model's ability to generalize to unseen data.

For example, the initial number of parameters for our model was 64 million, and it was reduced to 5.8 thousand parameters. Through several sessions of parameter tuning, such as adjusting the number of filters and layers, we were finally able to obtain similar results with less time and fewer parameters.

Comparison with Previous Work:
While previous studies have explored the use of deep learning for brain tumor classification, many existing models require intensive computations and may not be suitable for immediate diagnosis or deployment on edge devices.
Our lightweight CNN provides a more practical solution for resource-constrained environments, achieving similar accuracy at a significantly lower computational cost and training time compared to models like VGG16 and ResNet50.

Significance of Our Results:
The development of an accurate and lightweight brain tumor classification model has significant implications for clinical practice. It enables faster and more accessible diagnosis, particularly in resource-constrained environments where access to specialized equipment and expertise may be limited. This, in turn, enables earlier detection and treatment. The model's ability to generalize to unseen data indicates its potential for use in diverse clinical settings.

Limitations We Faced:
Although our model shows promising results, it is important to acknowledge some limitations. The dataset we used in this study was of medium to small size, despite our use of data augmentation. Further validation on larger and more diverse datasets is still needed.
Additionally, the lack of sufficient information, such as the number of parameters and training time, for lightweight models in previous scientific literature also represents a limitation, as the accuracy metric alone is not sufficient as a factor for comparison.

Future Research Directions:
Future research should focus on validating the model on larger and more diverse datasets. Moreover, exploring techniques to reduce the model size and computational complexity without compromising accuracy is a promising area for future research, from the model perspective.

From another perspective, interpreting the results of deep learning models used in brain tumor diagnosis via MRI is crucial; it goes beyond simply obtaining a diagnosis. It represents a cornerstone in building trust between physicians and these advanced technologies by clarifying the decision-making mechanisms within the complex models. It also enables the improvement of the performance of these models by detecting underlying biases and potential errors, ensuring diagnostic accuracy and reliability. Furthermore, interpretation contributes to enhancing patient safety by reducing diagnostic errors and determining responsibility in medical cases, in addition to opening new horizons for discovering biomarkers and understanding the nature of tumors more deeply. Finally, interpretation allows for the design of personalized treatments for each patient based on a precise understanding of the tumor's characteristics and potential response to treatment, making it an essential element for ensuring the safe and effective use of deep learning techniques in brain tumor diagnosis and improving the quality of healthcare provided to patients.

Our research also included an attempt to create a natural language processing model for images in order to analyze images that were identified as tumors, but we did not have enough time to complete this aspect of the research. Therefore, we hope to see future research in this direction.

General Conclusion

this thesis addressed the critical need for accurate and efficient brain tumor diagnosis by developing and evaluating a lightweight convolutional neural network (CNN) tailored for classifying brain tumors from MRI and CT images into (healthy / tumor).

Recognizing the limitations of traditional diagnostic methods and the computational demands of existing deep learning models, this research focused on creating a solution that balances high accuracy with practical deploy ability, particularly in resource-constrained environments.

Building upon a comprehensive review of existing literature, including machine learning approaches, novel CNN architectures, and deep transfer learning techniques, this work identified key areas for improvement and innovation. The success of architectures like VGG-16, AlexNet, and Xception, as highlighted in the related works, validated the potential for developing new architectures.

Furthermore, the literature review underscored the importance of transfer learning, benchmarking, and hybrid approaches in enhancing diagnostic accuracy.

The proposed methodology involved developing a custom CNN architecture and employing transfer learning techniques using pre-trained VGG16 and ResNet-50 models. Rigorous experimentation with various hyper-parameters, image processing techniques (including normalization, data augmentation, and resizing), and optimization strategies (such as early stopping and learning rate reduction) was conducted to optimize the model's performance.

The experimental results demonstrated the effectiveness of the proposed lightweight CNN, achieving competitive accuracy compared to more complex models while significantly reducing computational cost and training time. Specifically, the model achieved high accuracy on both MRI and CT images, demonstrating its ability to generalize across different types of medical images. The comparative analysis with VGG16, ResNet50, and an existing brain tumor classification model further highlighted the efficiency and effectiveness of our proposed model CNN.

The key contributions of this thesis include:

(1) the development of a CNN architecture specifically tailored for brain tumor classification from MRI/CT images

(2) the evaluation of the effectiveness of transfer learning in improving the performance of the CNN model with limited data

(3) a comparison of the performance of the proposed CNN with transfer learning and against existing techniques.

While the results are promising, it's important to acknowledge the limitations of this study. The dataset used was of medium size, and further validation on larger and more diverse datasets is needed. Additionally, the lack of sufficient information on lightweight models in previous literature posed a challenge for comprehensive comparison.

Future research directions should focus on validating the model on larger and more diverse datasets, exploring techniques to further reduce model size and complexity without compromising accuracy, and investigating methods for interpreting the results of deep learning models to enhance trust and understanding in clinical practice. Furthermore, exploring the potential of natural language processing models for image analysis represents a promising avenue for future research.

Finally, this thesis contributes to the field of brain tumor diagnosis by presenting a practical and effective lightweight CNN architecture that can potentially improve the speed, accuracy, and accessibility of diagnosis, ultimately leading to better patient outcomes.

References

[1] Gao, H., & Jiang, X. (2013). Progress on the diagnosis and evaluation of brain tumors. *Cancer Imaging: The Official Publication of the International Cancer Imaging Society*, *13*(4), 466–481. https://doi.org/10.1102/1470-7330.2013.0039

[2] Abd-Ellah, M. K., Awad, A. I., Khalaf, A. A. M., & Hamed, H. F. A. (2019). A review on brain tumor diagnosis from MRI images: Practical implications, key achievements, and lessons learned. Magnetic Resonance Imaging, 61, 300-318. https://doi.org/10.1016/j.mri.2019.05.028

[3] Nazir, M., Shakil, S., & Khurshid, K. (2021). Role of deep learning in brain tumor detection and classification (2015 to 2020): A review. Computerized Medical Imaging and Graphics, 91, 101940. https://doi.org/10.1016/j.compmedimag.2021.101940

[4] Roy, S., Nag, S., Maitra, I. K., & Bandyopadhyay, S. K. (2013). A review on automated brain tumor detection and segmentation from MRI of brain. arXiv.

[5] IU Health. (2024). Understanding brain tumor imaging: CT scan vs. MRI. IU Health Thrive. https://iuhealth.org/thrive/understanding-brain-tumor-imaging-ct-scan-vs-mri

[6] Deng, L. (2018). Artificial intelligence in the rising wave of deep learning: The historical path and future outlook [Perspectives]. IEEE Signal Processing Magazine, 35(1), 180–177.

[7] said bacheir
Applications-ML-Ch1
PDF

[8] A review on machine learning & it's algorithms. 2024.

[9] Omran, M. G., Engelbrecht, A. P., & Salman, A. (2007). An overview of clustering methods. Intelligent Data Analysis, 11(6), 583–605.

[10 Ahmad, N., & Nassif, A. B. (2022). Dimensionality reduction: Challenges and solutions. ITM Web of Conferences, 43, 01017.

[11] Aarthy, R. (n.d.). Introduction to images in computer vision. Scaler. https://www.scaler.com/topics/computer-vision-image

[12]
Magadza, T., & Viriri, S. (2021). Deep learning for brain tumor segmentation: A survey of state-of-the-art. Journal of Imaging, 7(2), 19. https://doi.org/10.3390/jimaging7020019

[13] Tjahyaningtijas, H. P. A. (2018). Brain tumor image segmentation in MRI image. IOP Conference Series: Materials Science and Engineering, 336(1), 012023.

[14 Elshaikh, B. G., Garelnabi, M. E. M., Omer, H., Sulieman, A., Habeeballa, B., & Tabeidi, R. A. (2021). Recognition of brain tumors in MRI images using texture analysis. Saudi Journal of Biological Sciences, 28(4), 2381-2387.
https://doi.org/10.1016/j.sjbs.2021.01.035

[15]
(Book)
Skansi, S. (2018). Introduction to deep learning. Springer.

[16] Vakalopoulou, M., Christodoulidis, S., Burgos, N., Colliot, O., & Lepetit, V. (2023). Deep learning: Basics and convolutional neural networks (CNN). HAL preprint.

[17] Sinha, R. K., Pandey, R., & Pattnaik, R. (2018). Deep learning for computer vision tasks: A review. arXiv.
https://doi.org/10.48550/arXiv.1804.03928

[18] Dounia Bouchekioua
Fault classification in photovoltaic systems using deep learning techniques
University of khasdi merbah ouargla
PDF

[19]
Srinivas, S. C., Sarvadevabhatla, R. K., Mopuri, K. R., Prabhu, N., Kruthiventi, S., & Babu, R. B. (2016). A taxonomy of deep convolutional neural nets for computer vision. Frontiers in Robotics and AI, 2, 36.

[20] Sinha, R. K., Pandey, R., & Pattnaik, R. (2017). Deep learning for computer vision tasks: A review. 2017 International Conference on Intelligent Computing and Control (I2C2).

[21] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research, 15(1), 1929–1958.

[22] Ng, A. Y. (2004). Feature selection, l1 vs. l2 regularization, and rotational invariance. Proceedings of the Twenty-First International Conference on Machine Learning, 78.

[23] Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. International Conference on Machine Learning, 448–456

[24] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems, 25.

[25] Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I, 818–833

[26] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv.

[27] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1–9.

[28] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 770–778.

[29] Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 4700–4708.

F. Sultana, A. Sufian, and P. Dutta. Advancements in image classification using convolutional neural network. In 2018 Fourth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN), pages 122–129. IEEE, 2018

[30] Olaoye, F., & Potter, K. (2024). Transfer learning and its role in machine learning. EasyChair Preprint no. 12570.

[31] Chatterjee, S., Alkhaldi, R., Yaadav, P., Bethineedi, D., Shreya, A., & Bankole, N. (2022). P.115 Diagnostic performance of machine learning based MR algorithm vs conventional MR images for predicting the likelihood of brain tumors. Canadian Journal of Neurological Sciences, 49(s1), S38. https://doi.org/10.1017/cjn.2022.207

[32]
Zia, R., Farhi, L., & Ali, Z. (2018). Performance analysis of machine learning classifiers for brain tumor MR images. Sir Syed University Research Journal of Engineering & Technology, 8(1), 32-41.

[33]
Farooqui, M. A. (2024). Machine learning in brain tumor diagnosis: Assessing the efficacy of diverse methods. Power System Technology, 48(4), 4315–4323. https://doi.org/10.52783/pst.1272

[34]
Ghosh, A., & Kole, A. (2021). A comparative study of enhanced machine learning algorithms for brain tumor detection and classification. 2021 International Conference on Computer Communication and Informatics (ICCCI).

[35]
Reddy, K. R., & Dhuli, R. (2023). A novel lightweight CNN architecture for the diagnosis of brain tumors using MR images. Diagnostics, 13(2), 312.
https://doi.org/10.3390/diagnostics13020312

[36]
Rai, H. M., Chatterjee, K., Gupta, A., & Dubey, A. (2020). A novel deep CNN model for classification of brain tumor from MR images. 2020 IEEE 1st International Conference for Convergence in Engineering (ICCE), 134-138.
https://doi.org/10.1109/ICCE50343.2020.9290740

[37]
Rai, H. M., Chatterjee, K., Gupta, A., & Dubey, A. (2020). A novel deep CNN model for classification of brain tumor from MR images. 2020 IEEE 1st International Conference for Convergence in Engineering (ICCE), 134-138.
https://doi.org/10.1109/ICCE50343.2020.9290740

[38]
Naseer, A., Yasir, T., Azhar, A., Shakeel, T., & Zafar, K. (2021). Computer-aided brain tumor diagnosis: Performance evaluation of deep learner CNN using augmented brain MRI. Computational Intelligence and Neuroscience, 2021, 5513500.
https://doi.org/10.1155/2021/5513500

[39]
Srinivas, C., et al. (2022). Deep transfer learning approaches in performance analysis of brain tumor classification using MRI images. Journal of Healthcare Engineering, 2022, 3381862.

[40]
Ahmad, S., et al. (2022). On the performance of deep transfer learning networks for brain tumor detection using MR images. IEEE Access, 10, 75869-75881.

[41]
Badjie, B., & Ülker, E. D. (2022). A deep transfer learning based architecture for brain tumor classification using MR images. Information Technology and Control, 51(2), 332-344.
http://dx.doi.org/10.5755/j01.itc.51.2.30835

[42]
Asif, S., Yi, W., Ain, Q. U., Hou, J., Yi, T., & Si, J. (2022). Improving effectiveness of different deep transfer learning-based models for detecting brain tumors from MR images. IEEE Access, 10, 34716-34730. https://doi.org/10.1109/ACCESS.2022.3153306

[43]
Asiri, A. A., et al. (2023). Brain tumor detection and classification using fine-tuned CNN with ResNet50 and U-Net model: A study on TCGA-LGG and TCIA dataset for MRI applications. Sensors, 23(1), 1.

[44]
Sarkar, A., Maniruzzaman, M., Alahe, M. A., & Ahmad, M. (2023). An effective and novel approach for brain tumor classification using AlexNet CNN feature extractor and multiple eminent machine learning classifiers in MRIs. Journal of Sensors, 2023, 5562725.

[45] Gimini-2.0-flash
Gemini Flash - Google DeepMind

[46] Brain tumor multimodal image (CT & MRI)
www.kaggle.com/datasets/murtozalikhon/brain-tumor-multimodal-image-ct-and-mri

[47] GeeksforGeeks . Evaluation metrics in machine learning.
https://www.geeksforgeeks.org/metrics-for-machine-learning-model