An Improved AlexNet Convolutional Neural Network Model for Brain Tumor Detection and Classification

Kofoworola Folakemi FAMUREWA
LCU/PG/002338

Being a PhD Thesis Submitted to the Department of Computer Science, Faculty of Natural & Applied Sciences, Lead City University, Ibadan, Oyo State, Nigeria

In Partial Fulfilment of the Requirements for the Award of Doctor of Philosophy Degree (PhD) in Computer and Information Science

2024

Certification

This is to certify that Kofoworola Folakemi FAMUREWA with matriculation number LCU/PG/002338 carried out this research work titled "An Improved AlexNet Convolutional Neural Network Model for Brain Tumor Detection and Classification" in the Department of Computer Science, Faculty of Natural and Applied Sciences, Lead City University, Ibadan, Oyo State, for the award of Doctor of Philosophy Degree (PhD) in Computer and Information Science and that this has not been previously submitted.

______________________ ______________________

Dr. Wilson Sakpere
(Supervisor)

Date

______________________ ______________________

Dr. Wilson Sakpere
(Head of Department)

Date

Dedication

This research work is dedicated to Almighty God for seeing me through the process of carrying out this research.

Acknowledgement

I give all glory and praise to Almighty God, the giver of power, wisdom, knowledge and understanding for His infinite grace to start and complete this research work, blessed be to His holy name.

I thank the management of Lead City University for providing an enabling environment for learning and carrying out project. I pray that God will increase the school on every side. I also wish to acknowledge the enormous contribution of the authors of the literature consulted, their wealth of knowledge helped me greatly in making this project a standard one.

I sincerely appreciate the effort of my supervisor Dr. W. Sakpere who despite his busy schedule took his precious time to direct, give instruction, correct and advise me. Thank you very much sir. I also appreciate the effort of the Postgraduate coordinator, Dr. A. A Waheed and all my lecturers within the department for impacting a great measure of knowledge unto me during the course of my study, may God bless you all.

My special appreciation goes to Mrs. F. O. Adelodun, Mr. A. Adeloye and Mrs O. Agbonavbare for their support academically. I pray that God will continue to bless you and you will not lack good things.

I would like to use this opportunity to thank my mother Mrs O. A. Famurewa and my loving and caring husband Mr. I. Ogedengbe for their spiritual, financial and physical support during the course of carrying out this research work. God will enrich you with heavenly and earthly blessings.

“Though the above-mentioned institutions and persons have assisted me in the process of this research work, I alone stand responsible for the errors, if any, found in the work.”

Abstract

Brain tumors are frequently categorized as malignant or benign. The treatment for brain tumors requires an early diagnosis and the usual method to detect brain tumor is Magnetic Resonance Imaging (MRI) scans. From the MRI scan, information about the abnormal tissue growth in the brain is identified. Human inspection, which may be time-consuming and not suitable for large number of MRI images, is the traditional method used in contemporary clinical routines for tumor detection and classification in MRI images. Recently, convolution neural networks (CNNs) have made imaging-based artificial intelligence solutions possible. When CNN models are applied on the MRI images, the prediction of brain tumor is done very fast and a higher accuracy helps in providing the treatment to the patient. These predictions also help the radiologist in making quick decisions. Even though CNNs has achieved great results in many tasks and domain, their sensitivity to input size remains a major problem that limits practical use cases. This work modified AlexNet CNN architecture to accept varying sizes of brain tumor images and then classify the tumor as cancerous or non-cancerous. The specific objectives were to acquire and preprocess MRI brain tumor images, develop CNN model that accept varying brain tumor images and evaluate the performance of the model. The implementation was done using Python and TensorFlow and it was executed on a desktop computer with Intel Core-i5 processor and 16 GB RAM. At the end of the training, the model achieved 89.86% training accuracy and 85.08% validation accuracy. An accuracy of 84.18% was achieved after assessing the model on test data. An evaluation of the model's performance revealed that this approach holds great potential.

Keywords: Artificial Intelligence, Brain Tumor, Convolutional Neural Network, Input Size Limitation, Magnetic Resonance Imaging.

Word Count: 274

Table of Contents

Content **Page**

List of Tables

List of Figures

List of Acronyms

Abbreviation	Meaning
AdaGrad:	Adaptive Gradient
ADAM:	Adaptive Moment Estimation
AEDs:	Anti-epilepsy drugs
AI:	Artificial intelligence
ANN:	Artificial Neural Networks
AUC:	Area under the ROC Curve
B2B:	Buckets to Buckets
BTS:	Back-Propagation through Structure
CAD:	Computer Aided Diagnosis
CNN:	Convolutional Neural Network
CNS:	Central Nervous System
CT:	Computerized Tomography
DDPG:	Deep Deterministic Policy Gradient
DICOM:	Digital Imaging and Communications in Medicine
DL:	Deep Learning
DNA:	Deoxyribonucleic Acid
DNN:	Deep Neural Network
DRL:	Deep Reinforcement Learning
DWT:	Discrete Wavelet Transform
EEG:	Electroencephalogram
EHO:	Elephant Herding Optimization
ENet:	Efficient Neural Network
FC:	Fully Connected
FCNN:	Fully Connected Neural Network

FDA:	Food and Drug Administration
fMRI:	Functional MRI
FNNs:	Feedforward Neural Networks
GAN:	Generative Adversarial Networks
GAP:	Global Average Pooling
GPU:	Graphics Processing Units
GRUs:	Gated Recurrent Units
HTML:	Hyper Text Markup Language
IoT:	Internet of Things
JPEG:	Joint Photographic Expert Group
JSON:	JavaScript Object Notation
KNN:	K-Nearest Neighbors
LIME:	Local Interpretable Model-Agnostic Explanation
LSTM:	Long Short-Term Memory
MB:	Megabyte
MB-SGD:	Mini Batch Stochastic Gradient Descent
MDT:	Multidisciplinary Team
ML:	Machine Learning
MLP:	Multilayer Perceptron
MR:	Magnetic Resonance
MRI:	Magnetic Resonance Imaging
MRS:	Magnetic Resonance Spectroscopy
NICE:	National Institute for Health and Care Excellence
NLP:	Natural Language Processing
NoSQL:	Not Only SQL
PCA:	Principal Component Analysis

PDF:	Portable Document Format
PDT:	Photodynamic Therapy
PET:	Positron Emission Tomography
PNN:	Probabilistic Neural Network
RBM:	Restricted Boltzmann Machine
ReLU:	Rectified Linear Unit
ResNet:	Residual Neural Network
RNNs:	Recurrent Neural Networks
RvNN:	Recursive Neural Networks
SGD:	Stochastic Gradient Descent
SHAP:	Shapley Additive Explanation
SOM:	Self-Organized Mapping
SPECT:	Single Photon Emission Computerized Tomography
SPP:	Space Pyramid Pooling
SQL:	Structured Query Language
SRS:	Stereotactic radiosurgery
SRT:	Stereotactic Radiation Therapy
SSAE:	Stacked Sparse Auto Encoder
SSGD:	Streaming Stochastic Gradient Descent
SVM:	Support Vector Machine
TKI:	Tyrosine Kinase Inhibitor
VEGF:	Anti-Vascular Endothelial Growth Factor
VGG:	Visual Geometric Group
VSC:	Variable Step Convolution
VSP:	Variable Step Pooling

WHO:	World Health Organization
XML:	Extensible Markup Language

List of Appendix

Chapter One

Introduction

1.1 Background to the Study

The uncontrolled and abnormal cell growth and division within the body is known as cancer[1]. In 2020, cancer accounted for approximately 10 million deaths worldwide, ranking second only to cardiovascular disease, according to the World Health Organization's (WHO) 2020 World Cancer Report[2]. The occurrence, as a mass or lump, of these abnormal cell growth and division in the brain tissue is called brain tumor[1].

Brain tumor is defined by the World Health Organization (WHO) as a form of tumor that affects the central nervous system. Generally speaking, abnormal brain cell formation leads to the formation of a brain tumor or intracranial neoplasm[3]. When a brain tumor develops, it can cause significant functional impairments to an individual and necessitates treatment strategies that strike a balance between the tumor's management and the brain's ability to function normally[4].

Brain tumors are frequently categorized as malignant (cancerous) or benign (non-cancerous)[5]. Benign brain tumors are the least aggressive kind of brain tumors. Benign brain tumors are slow-growing, unlikely to spread to other tissues, and arise from cells in the brain or its surrounding tissues. They also do not contain cancerous cells. They could get fairly big before they start to exhibit any symptoms that interfere with brain function. These tumors might not recur if they can be entirely removed. Nevertheless, depending on their size and position in relation to other brain regions, they may still induce serious neurological symptoms that necessitate immediate medical attention. A benign tumor may occasionally develop into a malignant one over time[4].

Malignant brain tumors have cancer cells and frequently lack distinct borders. Because they spread quickly and infiltrate surrounding brain tissue, they are considered to be potentially fatal. Brain cancer is another term for a malignant brain tumor. Malignant brain tumors can extend across the brain or to the spinal cord, but they seldom spread to other parts of the body like malignant tumors in other parts of the body do. Malignant brain tumors may relapse after treatment [4].

A brain tumor that originates in the brain is referred to as a primary brain tumor or cancer, regardless of whether it is malignant or benign. It may extend to other parts of the nervous system, but rarely extend to other areas of the body. Occasionally, tumors that begin in another part of the body can move through the bloodstream to the brain. This is referred to as a metastasis, or secondary cancer. Compared to primary brain tumors, these tumors are more frequent. Melanoma, lung, breast, kidney, colon, skin, and bowel cancers are the ones that have the highest chance of spreading to the brain[5]. Brain tumors originate from various cell types, form in different places, and could respond differently to treatment. If left untreated, all brain tumors have the potential to spread and harm normal brain tissue, which might be lethal or severely incapacitating[4].

Meningioma, glioma, and pituitary tumors are the three types of brain tumor that mostly occur depending on the affected location. The degree of malignancy varies among each type of these tumors[6]. Glioma is the most prevalent type of primary brain tumor. Gliomas are tumors that originate in the glial (neuroglia) cells that make up the brain's supporting tissue. Gliomas are classified according to the location and origin of the tumor, into many categories which are Astrocytoma, Ependymal, Glioblastoma, Oligoastrocytoma, and Oligodendroglioma[4, 5].

Every tumor is categorized according to a standard established by the World Health Organization (WHO). Based on the location and type of tumor cells, the WHO has classified over 120 different types of brain tumors, making this an extremely difficult diagnosis. A number between 1 and 4, typically represented by the Roman numerals I through IV, is assigned to tumors based on the cells in which they originate. This quantity, known as the "grade," indicates how quickly the cells can proliferate and are likely to spread. For the purposes of treatment planning and result prediction, this rating provides essential information. Grades I and II are lower grade cancers that are often associated with long-term survival, whereas grades III and IV are higher grade tumors that grow more quickly and can cause more damage to the brain. Higher grade tumors are considered malignant or cancerous[4].

Globally, the incidence of all central nervous system tumors is approximately 4.0 per 100,000 people; it varies according to age, gender, race, and location, with Northern Europe having the highest frequency, followed by Australia, the US, and Canada. Meningioma is the most frequent tumor, making up almost 37% of all tumors; glioma, on the other hand, is the most prevalent malignant tumor, making up 75.0% of malignant tumors in the central nervous system and having an annual incidence of six cases per 100,000 persons[7]. There have reportedly been some discrepancies in the risk factors for brain tumors compared to other types of tumors[8]. Despite the discrepancies, some risk factors linked to the development of brain tumors include prior exposure to high doses of ionizing radiation, such as X-rays, CT scans, and MRIs, particularly for therapeutic purposes.

According to reports, children who receive radiation therapy for different medical conditions are susceptible to brain tumors approximately 15 years after the radiation treatment. Development of radiation-induced brain tumors is very prevalent in children with leukemia. Conflicting information has been published in publications about the role of various risk factors, such as cell phones, viruses, allergies, alcohol, N-nitroso compounds, infections, chemicals, and smoking, in the development of brain tumors. While there isn't a clear statement on cell phones as risk factors, the World Health Organization (WHO) recommends limiting mobile phone use because they may be harmful[9].

With the growth of brain tumor, whether cancerous or non-cancerous, the pressure inside the brain might increase. This pressure can damage the brain and cause the death of the person[10]. The techniques for treating brain tumors include steroids, surgery, chemotherapy, radiation, radiosurgery, and proton beam therapy. These techniques can be used in combination[11].

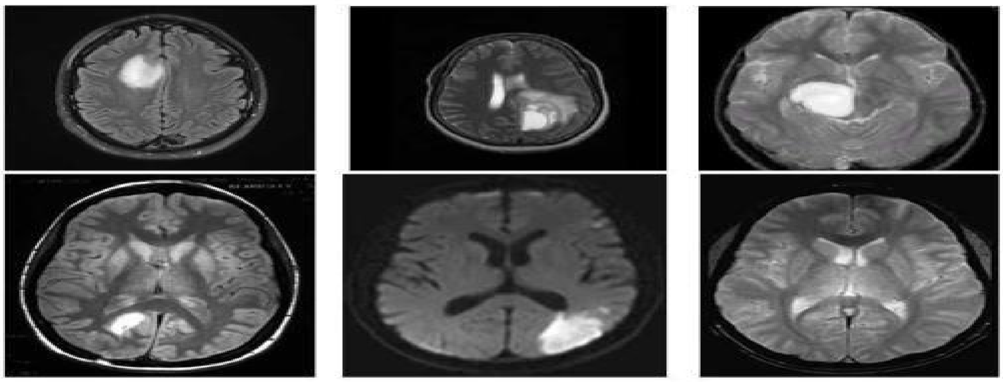


Figure 1.1: Brain Tumor Images[10]

Improving the chances of a successful course of treatment for brain tumors requires an early diagnosis. The shape, size, location, and metabolism of brain tumors can generally be determined with the use of a variety of medical imaging techniques, including magnetic resonance spectroscopy (MRS), magnetic resonance imaging (MRI), Positron Emission Tomography (PET), Computed Tomography (CT), Single-Photon Emission Computed Tomography (SPECT), and Magnetic Resonance Imaging (MRI) [1].

Although these techniques are utilized in combination to provide the most thorough information regarding brain malignancies, MRI is still the gold standard because of its excellent soft tissue contrast and widespread availability. MRI is a non-invasive in vivo imaging technique that uses radio frequency signals to stimulate target tissues to create their internal images under the influence of a magnetic field. By adjusting excitation and repetition durations during image capture, images of various MRI sequences can be produced These various MRI modalities generate different types of tissue contrast images, therefore providing valuable structural information and allowing diagnosis and segmentation of tumors along with their subregions[12]. MRI image provides more details about given medical imaging than the CT or ultrasound image. An MRI scan can provide specific details about the anatomy of the brain and identify any abnormalities in the brain tissue[13].

Human inspection is the traditional method used in contemporary clinical routines for tumor detection, segmentation, and classification in MRI images. Medical professionals and radiologists may find it laborious and time-consuming to visually interpret brain tumors in MRI images. In addition, this technique is not suitable for large number of multimodal MRI images[14].

This approach often necessitates a basic microscope and could result in erroneous classification[10]. Furthermore, operator intervention-related noise in the MRI can result in incorrect classification. As a result, development of robust automated system is required as they are more cost-effective. Automated detection of tumor in MRI images is essential as high accuracy is needed when dealing with human life[14].

One of the amazing attempts in utilizing Artificial Intelligence in identifying brain tumor in MRI images is deep learning (DL) architecture[15]. DL is also used in various medical imaging applications and Computer Aided Diagnosis (CAD) systems, demonstrating the enormous potential of these approaches in automating diagnosis. Convolutional neural network (CNN), on the other hand, is one of the other approaches that have demonstrated consistent efficacy in the classification of radiological images over the past few decades[16]. Deep learning algorithms of this kind are used in many different applications, one of which is pattern recognition[17].

Conventional machine learning techniques approach issues by decomposing them into smaller components, including mathematical modeling, data collection and cleansing, feature extraction and selection, training, optimization, and model performance assessment. This lengthy process of modeling problems with this kind of methodology increases to the computational cost and simulation time for large-scale problems. In contrast to this conventional approach, deep neural networks (DNNs) are end-to-end learning techniques that can reduce the cost of creating and extracting complex features by combining multiple processing steps into a single DNN. To achieve precise and competitive performance, DNN only requires enough data and proper alteration of its hyper-parameters[18].

Deep learning (DL) based on convolution neural networks (CNNs) has made imaging-based artificial intelligence (AI) solutions possible. The goal of DL-guided solutions is to support clinical decision making[7]. CNN is the latest development in the field of machine learning which is used in the diagnosis of diseases based on medical imaging, specifically CT and MRI images. Because CNN does not require preprocessing or feature extraction before the training process, it has been increasingly popular in the classification and grading of medical images in recent years[19].

Generally speaking, CNNs are made to handle raw pictures and are intended to reduce or eliminate some data pre-processing stages. The input layer, convolution layer, RELU layer, fully connected layer, classification layer, and output layer are the several layers that make up CNN. CNN primarily consists of and depends upon two processes: the convolution, which is carried out using trainable filters with predefined specifications that are adjusted during the training phase and down sampling[20].

Classifying brain MRI images into two basic categories—normal and abnormal—and grading the abnormal brain MRI images into multiple forms of brain cancer are the two general categories into which the deep learning applications for brain tumor classification can be split. The CNN has been used for brain tumor detection and grading, and it attracts researchers' attention as a potent tool in the field of disease detection and classification. This will improve the accuracy of the brain tumor grading that has been detected, assist doctors in formulating the best course of treatment, and ultimately raise the healing percentage[6]. Researchers have shown the immense potential of imaging techniques to lessen the burden that medical experts bear. It also makes it possible to provide more assistance with patient care, lessen burnout, and lower patients' overall medical expenses[21].

1.2 Statement of the Problem

Various convolutional neural networks models have been developed for the purpose of detecting and classifying brain tumors. Additionally, a lot of researchers have improved performance and shortened computation times by retraining pre-trained CNN models. The issue with nearly all of these current networks is that their input sizes are fixed[22, 23]. As a result, the majority of CNNs now have a limit in training and prediction[24]. CNNs can attain state-of-the-art performance in various tasks and domains, but their intrinsic sensitivity to image size limits the practical use cases[25]. Furthermore, networks that are trained on a certain image size perform poorly on other image sizes at evaluation[26].

Additionally, since consistency between input shapes is a necessary condition for typical Neural Network architectures and a really helpful condition for successfully training them, image recognition datasets are made up of a set of images that have been preprocessed to have exactly the same width and height. As a result, image recognition datasets are usually resized to sizes that match the input size requirement of the CNN architecture[27].

Brain tumor images come in different sizes. Any attempt to resize these images in order to meet the input size specification of a particular CNN model might result to a loss of vital information that could affect the accuracy and reliability of the brain tumor image classification model.

1.3 Aim and Objectives of the Study

The aim of this work is to develop an improved AlexNet convolutional neural network model for brain tumor detection and classification.

The specific objectives are to:

1. Pre-process the acquired MRI brain tumor images.
2. Implement a CNN model that will accept varying sizes of MRI brain tumor images and then classify the brain tumor as cancerous and non-cancerous.
3. Evaluate the performance of the model in terms of accuracy, precision, recall and f1 score.

1.4 Research Questions

1. How should the MRI brain tumor images acquired be preprocessed?

2. How can the CNN model that will accept varying sizes of MRI brain tumor images and then classify the brain tumor be implemented?
3. How should the model's performance be tested?

1.5 Significance of the Study

The intention of designing this new CNN model is that the model will be able to accept varying sizes of brain tumor MRI images as input and learn effectively. The model will help physicians and radiologists identify brain tumors accurately, prevent misdiagnosis of brain tumor, ease doctors and radiologists stress, increase their efficiency, decrease patient waiting time in hospital and ultimately improve patient care. This will have a huge positive impact on the healthcare industry.

1.6 Scope of the Study

In order to achieve the objectives of the study, brain tumor MRI images were collected from different directories on Kaggle website. The images were preprocessed and the model was developed and trained. Confusion matrix was used as the tool to evaluate the performance of the model. From the result obtained from the confusion matrix, the accuracy, precision, recall and F1-score were calculated. The implementation of this model was done using Python and TensorFlow and it is executed on a desktop computer with Intel Core-i5 processor and 16 GB RAM.

1.7 Limitations of the Study

The limitations encountered in the course of carrying out this study include:

1. There is paucity of literature on CNN models that accept varying image size.
2. Limited publicly available brain tumor image dataset with large image sizes.
3. Difficulty in collecting brain tumor MRI images from private domain.

1.8 Operational Definition of Terms

i. Artificial Intelligence: Artificial intelligence is a broad term that refers to techniques that allow computers to simulate human behavior.

ii. Brain Tumor: A brain tumor or intracranial neoplasm is a form of tumor that affects the central nervous system.

iii. Cancer: Cancer can be defined as the uncontrolled and abnormal cell growth and division within the body.

iv. Convolutional Neural Network (CNN): A Convolutional neural network is a neural network that has one or more convolutional layers and is used principally for image processing, classification, segmentation.

v. Deep Learning (DL): Deep learning is a branch of artificial intelligence (AI) that is used to mimic the function of the human brain particularly, pattern recognition by passing input via various layers of the neural network.

vi. Machine Learning: Machine Learning (ML) is a branch of artificial intelligence (AI) that allows computers to learn from data in a manner akin to that of the human brain without the need for explicit programming.

vii. Magnetic Resonance Imaging (MRI): is a non-invasive in vivo imaging technique that uses radio frequency signals to stimulate target tissues to create their internal images under the influence of a magnetic field.

viii. Tumor: A tumor is a swelling or lump that develops when certain cells grow and proliferate abnormally.

Endnotes

1. I. Ali, D. Cem, & Ş. Melike. *Review of MRI-based Brain Tumor Image Segmentation Using Deep Learning Methods*. **Procedia Computer Science.** 102 (2016), 2016, 317 – 324.

2. H. Sung, J. Ferlay, R. L. Siegel, M. Laversanne, I. Soerjomataram, & Jemal, A. *Global Cancer Statistics 2020: GLOBOCAN Estimates of Incidence and Mortality Worldwide for 36 Cancers in 185 Countries.* **Cancer J. Clin.** 71, 2021, 209–249.

3. Open Access Publisher. *Brain Tumor*. 2022, https://www.omicsonline.org/open-access-journals-list.php.

4. Cancer Support Community. *Frankly Speaking about Brain Tumors.* www.cancersupportcommunity.org 2013, 1-96

5. Cancer Council. *Understanding Brain Tumors: A Guide for People with Brain or Spinal Cord Tumors, their Families and Friends*. 2020. May 2020 edition, first published 1995, Next Ed. 2022.

6. M.A. Ali, A. Hiam, A.Q. Isam, A. Amin, & A. Wafaa. *Brain Tumor Classification Using Deep Learning Technique - A Comparison between Cropped, Uncropped, and Segmented Lesion Images with Different Sizes*. **International Journal of Advanced Trends in Computer Science and Engineering.** 8(6), 2019, 3684– 3691.

7. L. Gaur, M. Bhandari, T. Razdan, S. Mallik, & Z. Zhao. *Explanation-Driven Deep Learning Model for Prediction of Brain Tumour Status Using MRI Image Data*. **Front. Genet.** 13:822666. 2022.

8. K.A.A. Abolanle, S. Amina, A. Muhammad, A. Hina, & K.T. Omowumi. *Brain Tumor: An overview of the basic clinical manifestations and treatment.* **Glob J Cancer Ther.** 6(1), 2020, 038-041.

9. Q.T. Ostrom, & J.S. Barnholtz-Sloan. *Current state of our knowledge on brain tumor epidemiology.* Current Neurology and Neuroscience Reports 11: 2011, 329-335. Link: https://bit.ly/2T5lI7V

10. H.U. Rehman, S. Masood, & M.A. Khan. *Brain Tumor Classification using Deep Learning Methods.* **International Journal of Information Systems and Computer Technologies (IJISCT).** Vol. 1(1), 2022, 37-53.

11. Brain & Spine Foundation. *A guide for patients and carers*. www.brainandspine.org.uk 2020, 1-56.

12. A. Drevelegas, & N. Papanikolou, *Imaging modalities in brain tumors Imaging of Brain Tumors with Histological Correlations*. Berlin: Springer. 2011, Chapter 2, 13-34.

13. J. Seetha, & S.S. Raja. *Brain Tumor Classification Using Convolutional Neural Networks*. **Biomed. & Pharmacol. J.** 11(3), 2018, 1457-1461.

14. k. Sharma, A. Kaur, & S. Gujral. *Brain Tumor Detection based on Machine Learning Algorithms*. **International Journal of Computer Applications (0975 – 8887).** (103)1, 2014, 7-11

15. M. K. Abd-ellah, A. I. Awad, A. M. Ashraf, & H. F. A. Hamed. *A Review on Brain Tumor Diagnosis from MRI Images: Practical Implications, Key Achievements, and Lessons Learned*. **Magnetic Resonance Imaging.** 61, 2019, 300–318.

16. N. Ghassemi, A. Shoeibi, & M. Rouhani. *Deep Neural Network with Generative Adversarial Networks Pre-Training for Brain Tumor Classification Based on MR Images*. **Biomedical Signal Processing and Control.** 57, 2020, 101678.

17. G. Litjens, T. Kooi, B.E. Bejnordi, A.A.A. Setio, F. Ciompi, M. Ghafoorian, J.A.W.M. Laak, B. Ginneken, & C.I. Sánchez. *A survey on deep learning in medical image analysis*. **Medical Image Analysis.** 42, 2017, 60–88.

18. Ö. F. Ertuğrul. *A novel clustering method built on random weight artificial neural networks and differential evolution*. **Soft Computing.** 24, 2020, 12067–12078.

19. L. Eyal, J. Rajan, R. Narges, M.F. Girish, & W.L. Yvonne. *State of the art: Machine learning applications in glioma imaging*. **American Journal of Roentgenology**, 212(1), 2019, 26-37.

20. M. Neha, K.M. Yogesh, M. Shruti, & M. Divya. *Detection of Brain Tumor in MRI Image through Fuzzy-Based Approach*. High-Resolution Neuroimaging-Basic Physical Principles and Clinical Applications, IntechOpen, 2018.

21. S. Wojciech, W. Thomas, & M.K. Robert *Explainable Artificial Intelligence: Understanding, Visualizing and Interpreting Deep Learning Models*. **arXiv:1708.08296 [cs.AI],** 2017.

22. A. Abhishek, R. Aditya, S. Aravind, C.R. Naveen & P.L. Deepa. *Brain Tumor detection from brain MRI using Deep Learning*. **International Journal of Innovative Research in Applied Sciences and Engineering (IJIRASE)** 3(2), 2019, 458-465.

23. A. Laith, Z. Jinglan, J.H. Amjad, A. Ayad, D. Ye, A. Omran, J. Santamaría, A.F. Mohammed, A. Muthana, & F. Laith. *Review of Deep Learning: Concepts, CNN Architectures, Challenges, Applications, Future Directions*. **Journal of Big Data**. 2021, 1–74.

24. K. He, X. Zhang, S. Ren, & J. Sun. *Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition.* **[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence**, 37(9), 2015, 1904-1916.

25. H. Elad, W. Berry, H. Itay, B. Tal, H. Torsten, & S. Daniel. *Mix & Match: Training Convnets with Mixed Image Sizes for Improved Accuracy, Speed and Scale Resiliency*. **arXiv.** 2019, 1-13.

26. T. Hugo, V. Andrea, D. Matthijs, & J. Herv´e. *Fixing the train-test resolution discrepancy*. **CoRR**, abs/1906.06423, 2019.

27. P. Ferran, G. Dario, & L. Jesus. *Training CNNs using high-resolution images of variable shape.* **6th BSC Severo Ochoa Doctoral Symposium.** 2023, 60-61.

Chapter Two

Literature Review

2.1 Conceptual Review

2.1.1 Overview of Brain Tumor

The building blocks of human bodies are called cells, and each cell has it own unique structure and function[1]. Cells are composed of tissues and organs. The body produces new cells continuously to support growth, replace deteriorated tissue, and repair wounds[2]. Cells proliferate and expand organically as we mature from childhood to adulthood and as our bodies heal wounds and restore damaged tissue[1].

Normally, cells reproduce and die in an orderly manner, so that each new cell replaces the lost cell. However, occasionally, cells develop aberrantly and continue to expand. The aberrant cells aggregate into a lump or mass known as a tumor[2]. A tumor is a growth or lump that develops when specific cells proliferate and expand improperly[1]. An aberrant cell development is called a tumor[2].

The brain is a complex, elegant, and sophisticated collection of nerve cells and tissue. It effortlessly governs our senses (sight, hearing, taste, smell, and touch), our personality (thoughts, memory, intelligence, speech, judgment, and emotions), our blood pressure, heart rate, breathing, and other vital bodily functions, and our movement (coordination, balance, and negotiating space) in our surroundings. When aberrant brain cells proliferate to form a tumor, it can impair our ability to function normally and necessitates treatment options that strike a balance between the tumor's needs and our brain's ability to function[3].

A brain tumor is an abnormal development of tissue or cells in the brain or spinal cord that can impair normal brain function[3]. A brain tumor or intracranial neoplasm occurs when aberrant cells form within the brain[4]. Brain tumors are commonly categorized as either benign (non-cancerous) or malignant (cancerous). Tumors in different bodily sections are also referred to by these terms[2].

A benign brain tumor is the term used to describe the least aggressive kind of brain tumor. Benign brain tumors grow slowly, are unlikely to spread to other tissues, and arise from cells in the brain or its surrounding tissues. They also do not contain cancerous cells. Before exhibiting any symptoms that alter brain function, they could grow to be rather enormous. These tumors might not recur if they can be entirely removed. Nevertheless, depending on their size and proximity to other brain regions, they may still induce serious neurological symptoms that necessitate immediate medical attention. A benign tumor may occasionally develop into a malignant one over time[3].

Malignant brain tumors contain cancer cells and often lack distinct borders. Because they spread quickly and infiltrate adjacent brain tissue, they are regarded as potentially fatal. Brain cancer is another term for a malignant brain tumor. Malignant brain tumors can extend across the brain or to the spinal cord, but they seldom spread to other parts of the body like malignant tumors in other sections of the body do. Even after treatment, these cancers might reappear[3].

Primary brain tumor/cancer refers to a brain tumor that originally appears in the brain, regardless of whether it is malignant or benign. Rarely does it extend to other areas of the body, yet it might affect other areas of the nervous system. Tumors can occasionally begin in one area of the body and move through the bloodstream to the brain. This is referred to as a metastasis, or secondary cancer. Compared to primary brain tumors, these tumors are more frequent. The most common cancers that spread to the brain include melanoma, lung, breast, kidney, colon, skin, and bowel cancers. A metastasis retains the original cancer's name. For instance, although the patient may be experiencing symptoms due to the cancer's presence in the brain, the condition is still referred to be metastatic bowel cancer after it has progressed to the brain. Treatment for metastases depends on where they first appeared[2, 3]. If ignored, all brain tumors have the potential to spread and cause harm to healthy brain tissue, which might be deadly or very debilitating. Each person's experience with brain and spinal cord tumors varies. They originate from various cell types, form in different places, and could respond differently to treatment[3].

2.1.2 Types of Brain Tumor

The brain is composed of various tissues and cells some of which have the potential to transform into various kinds of tumors. There are more than 40 forms of primary brain and spinal cord cancers (sometimes called central nervous system or CNS tumors). They may originate from any region of the spinal cord or brain. Tumors are categorized according to the kind of cell they originate in and the expected behavior of those cells (based on their genetic makeup)[2]. Primary brain tumor can group into two classes which are Glioma tumors and non-glioma tumors.

2.1.2.1 Glioma Tumors

Glioma tumors are the most prevalent type of primary brain tumor. Tumors known as gliomas originate from the glial (neuroglia) cells that are present in the brain's supporting tissue. Gliomas are classified according to where they occur and where the tumor first appears. These comprise glioblastoma, oligoastrocytoma, oligodendroglioma, ependymal, and astrocytoma.

2.1.2.2 Non-Glioma Tumors

Meningioma, pituitary tumor, schwannoma, and medulloblastoma are examples of non-glioma tumors. Central Nervous System (CNS) lymphoma is another kind of primary brain tumor. CNS Lymphoma is a cancerous primary brain tumor that develops from the lymphocytes found in the eyes, brain or spinal cord. It usually stays inside the central nervous system. Chemotherapy and/or radiation therapy are frequently used as treatments[3, 2].

2.1.3 Grading Tumors

Every tumor is categorized according to a standard established by the World Health Organization (WHO). Based on the location and type of tumor cells, the WHO has classified over 120 different types of brain tumors, making this an extremely difficult diagnosis. Tumors are given a name based on the cells where they originate, and a digit ranging from 1–4, usually represented by Roman numerals I-IV. The "grade" is a numerical value that indicates how quickly the cells can proliferate and are likely to spread. When it comes to treatment planning and outcome prediction, this grading information is essential. Higher grade tumors (grades III & IV) grow more quickly, can cause more damage, and are frequently more difficult to cure than lower grade cancers (grades I & II), which are typically associated with long-term survival and less aggressiveness. These are regarded as carcinogenic or malignant[3].

2.1.4 Causes and Risk Factor of Brain Tumor

It is uncertain what exactly causes the majority of brain and spinal cord tumors and why some are malignant while others are not. Known risk-raising factors include radiation therapy and family history[2]. Individuals who received radiation therapy as children for leukemia and pediatric cancers have been found to be at risk for acquiring brain tumors in their adult years. Nonetheless, the danger is minimal[1]. Additionally, it was often believed that certain brain cancers developed as a result of head trauma, but this is now shown to be untrue except from of a very tenuous connection between certain brain trauma and meningiomas (tumors that grow on the layers of membrane that surround the brain)[1]. There have been worries that brain tumor development is influenced by electromagnetic radiation from microwave ovens and cell phones. There is currently no proof linking cell phone use to cancer. Research on the possible long-term consequences of cell phone use is still ongoing. There is no proof that properly functioning ovens emit electromagnetic radiation at levels that are dangerous to humans[2].

When comparing the risk factors of brain tumors to those of other body tumors, there have been some discrepancies. Even though the risk factors are controversial, some risk factors linked to the formation of brain tumors include prior exposure of the head to high doses of ionizing radiation, such as X-rays, CT scans, and MRIs, especially for therapeutic purposes. Children who receive radiation therapy for a variety of illnesses have been shown to be at risk for brain tumors approximately 15 years after the radiation[5]. Development of radiation-induced brain tumors is very prevalent among children with leukemia. Conflicting information has been published in publications about the role of various risk factors, such as cell phones, viruses, allergies, alcohol, N-nitroso compounds, infections, chemicals, and smoking, in the development of brain tumors. While there isn't a clear declaration available about cell phones as risk factors[6], the World Health Organization (WHO) recommends limited use of mobile phones because they may be harmful.

The symptoms of brain and spinal cord tumors do not appear beforehand. Nonmalignant tumors grow extremely slowly over a period of years and can develop to significant sizes before being discovered. Malignant brain tumors grow more swiftly and typically have been present for a shorter period of time when they are detected. The time a tumor has been present or its malignancy cannot be determined by the emergence of symptoms[1].

2.1.5 Incidence and Prevalence of Brain Tumors

The charity Brain Tumor Research estimates that 16,000 people in the UK receive a brain tumor diagnosis annually. Additionally, it's estimated that approximately 60,000 people in England are affected with brain tumors. People of all ages are susceptible to primary brain tumors. Although they are more common in adults between the ages of 50 and 70, they are also the most prevalent malignancy in youngsters, second only to leukemia[1].

In Australia, an estimated 2000 cases of malignant brain tumors are diagnosed annually. An estimated 100 children between the ages of 0 and 14 receive a diagnosis every year. Malignant tumors are less common than benign brain and spinal cord tumors. Collectively, Victoria, Queensland, and Western Australia had over a thousand benign brain and spinal cord tumors in 2015[2]. Furthermore, there are a growing number of individuals receiving a diagnosis of metastases, or secondary tumors. This is because there is a greater likelihood of secondary tumor development because more people are now surviving other cancers, like breast cancer, thanks to improved detection and treatments[1].

2.1.6 Symptoms of Brain Tumors

Sleepiness, memory issues, vision issues, speech issues, numbness or weakness on one side of the body, seizures, headaches, nausea, and vomiting are the primary symptoms of a brain tumor. Hormonal fluctuations and psychological changes are also possible in certain individuals. The symptoms are primarily determined by the location of the tumor in your brain and the areas of it that are affected. Certain symptoms stem from the tumor exerting more force on the brain inside the skull[1]. Instead of happening all at once, symptoms may develop gradually over time.

i. Sleepiness: Drowsiness and sleepiness may be the result of increased pressure on the brain caused by a brain tumor. It is possible to feel sleepy throughout the day and to sleep longer at night than usual. A person may experience difficulty focusing and feeling as though he lacks clarity of mind.
ii. Memory Issues: Long-term memory issues, short-term memory issues, and general amnesia can all be symptoms of brain tumor-related memory impairment. One may discover that while he can recall events from a few years ago, but have trouble recalling recent information, such as the name of a new acquaintance or phone number.
iii. Visual Issues: Someone could experience blurred vision, double vision, difficulties focusing, or issues with one side of the vision (one could bump into objects to the left or right because he can't see them there).
iv. Speech Problems: Speaking difficulties brought on by a brain tumor might cause a person to slur words or, more frequently, mix and jumble words. The word "dysphasia" is the term used to describe this word jumble and mixing. Dysphasia patients frequently understand what is being said to them and know what they want to say, yet when they try to speak, their words often sound incorrect. Knowing that the terms are incorrect, the person could feel angry and frustrated. The first signs of dysphasia are typically trouble finding the correct word or stumbling over simple words. While this is something that we all occasionally do, speech issues caused by brain tumors can worsen over time

and eventually render a person incapable of communicating. These individuals may also experience difficulties comprehending spoken language.

v. Physical Problems on One Side of the Body: Physical issues resulting from a brain tumor may include tingling, numbness, weakness, or impaired balance and coordination on one side of the body, usually the arms or legs. Although the symptoms may come and go, most patients discover that in the days and weeks that follow their initial appearance, their medical issues worsen.

vi. Hormonal Changes: The production of hormones in the brain may be influenced by tumors which can result to changes in hormones. Someone may notice a decrease in desire for sex or a decline interest in having sex. Men may become impotent and women may notice a cessation of their menstrual cycles.

vii. Seizures: Unusual electrical activity in the brain caused by a brain tumor may result in seizures (fits). When a person experiences multiple seizures, epilepsy is diagnosed. There are numerous varieties of seizures, and every individual will have a unique experience. Periodic absences and prolonged seizures that cause convulsions and unconsciousness are two different types of seizures. The specific area of the brain affected and the degree of abnormal electrical activity will determine the kind, duration, and intensity of the seizure[1, 2, 3].

viii. Personality Changes: A brain tumor may result in alterations to a person's behavior and personality. For the affected person as well as their friends and family, these changes may be quite distressing. It's possible to amplify already-existing personality traits or features or to produce new ones. Previously calm individuals may exhibit aggressive, violent, and easily agitated behavior. They may yell and use profanity. They can exhibit a loss of inhibition and act differently than they typically would. Dealing with personality and behavior changes may be challenging, particularly when the individual going through them isn't always conscious of how they're affecting their friends and family.

ix. Headache: Headaches that become more intense over time are common in people with brain tumors. Tumors rarely cause headaches on their own. It is far more likely to be associated with stress, headaches, colds, the flu, or dehydration. Nausea and vomiting may accompany headache, but these are also potential migraine symptoms. Headaches should be treated more carefully and a doctor should be consulted if they coexist with any of the other primary signs of a brain tumor[1, 2, 3].

2.1.7 Diagnosing Brain Tumor

While some people have no apparent symptoms, others may have symptoms that point to the presence of a brain tumor. Long-term headaches, seizures or convulsions, trouble speaking or thinking, personality changes, tingling or stiffness in one side of the body, loss of balance, altered eyesight, nausea, and/or disorientation are common symptoms. If a patient has these symptoms, a physician will inquire about the patient's general health and medical history. They will also recommend a range of diagnostic tests to identify the underlying cause of the issues and look for solutions[3].

A physician will assess the nervous system, mental and physical alertness, and typical brain functions such as reflexes, judgment, smell, and taste as part of the diagnostic tests. This is known as neurological examination. In the event that the patient's answers are abnormal, more testing may be recommended by a neurologist or neurosurgical oncologist, or a brain scan may be ordered[3]. Additionally, blood tests may be performed on the patient to assess their general health. Additionally, blood tests can be conducted to determine whether the tumor is generating hormones at unusually high levels, which may indicate a problem with the pituitary gland[2].

2.1.8 Brain Tumor Scanning and Imaging Techniques

To determine whether a brain tumor is present and to pinpoint its precise growth location, a scan is the first step. A scan looks at the brain and spinal cord from several perspectives to produce computerized images of them. A contrast agent, sometimes known as a dye, is sometimes used in scans to help the physician distinguish between normal and diseased tissue. Depending on the nature and location of the tumor, a patient may require more than one type of scan to make the diagnosis[3].

2.1.8.1 Computerized Tomography (CT) Scan

A CT scan is a unique kind of X-ray that uses multiple angles to capture images of the brain. The procedure involves having the patient lie on a table with a scanner rotating around his head. The examination is short and painless. Occasionally, a dye (CT contrast agent) is injected into an arm vein to enhance the visibility of the tumor in the images[1].

2.1.8.2 Magnetic Resonance Imaging (MRI) Scan

Strong magnetic fields and radio waves are used in magnetic resonance imaging (MRI) scans to create images of the head and brain. It creates extremely detailed photos, which sets it apart from a typical X-ray. The patient will lie in a lengthy tube for the duration of the scan. While the scan is painless, the scanner makes a lot of noise. There are headphones or earplugs (sometimes both) available. The patient may converse with hospital professionals while he is in the scanner via to the headphones. In order to make the tumor appear larger in the images, an MRI contrast agent (dye) may occasionally be injected into an arm vein. An MRI scan is frequently performed following a preliminary CT scan[1].

2.1.8.3 Magnetic Resonance Spectroscopy (MRI Spect or MRS) Scan

An advanced form of magnetic resonance imaging is called a magnetic resonance spectroscopy (MRS or MRI Spect) scan. It calculates the body's metabolite levels. An MRS can identify abnormal patterns of activity to assess a tumor's aggressiveness, gauge its response to treatment, and identify the type of tumor[3]. It is possible to perform it concurrently with a typical MRI[2].

2.1.8.4 MR (Magnetic Resonance) Tractography Scan

A magnetic resonance (MR) tractography scan can be used to visualize the brain's message channels, or "tracts," such as the visual tracts. It can aid in the planning of glioma treatment[2].

2.1.8.5 MR Perfusion Scan

This kind of scan displays the quantity of blood going to different areas of the brain[2]. It is useful in determining the aggressiveness and grade of tumors as well as distinguishing between dead tumor tissues and recurring tumors[3].

2.1.8.6 Functional MRI (fMRI)

When patients carry out tasks, functional MRI monitors how much oxygen and blood is used in the brain. The motor, sensory, visual, and linguistic areas of the brain can be identified with an fMRI, which aids doctor in carefully planning the surgical procedure[3].

2.1.8.7 Positron Emission Tomography (PET) Scan

A radioactive material is used in positron emission tomography (PET) scans to visualize hypermetabolic activity, such as that found in malignant cells, anomalies from tumors, or scar tissue. Additionally, PET is utilized in brain mapping techniques[3]. PET scan produces comprehensive three-dimensional colour images of the brain. Some individuals may also get this kind of scan after receiving an MRI or CT scan[1].

2.1.8.8 SPET or SPECT (Single Photon Emission Computerized Tomography) Scan

A SPECT scan reveals how blood flow the brain. After injecting a little dose of radioactive fluid to the patient, a specialized camera will scan the patient's brain. Regions with increased blood flow, such a tumor, may appear brighter on the scan[2].

2.1.8.9 Spinal Tap (also called a Lumbar Puncture)

A spinal tap involves using a needle to collect a sample of cerebrospinal fluid from the spinal column. The fluid is examined for the presence of cancer cells in a lab[2].

2.1.8.10 EEG (Electroencephalogram)

An EEG may be ordered if the patients have a seizure as their initial symptom. An EEG requires attaching wires to the head for 20 to 30 minutes, during which a recording of the brain's electrical activity is made. The patient's scalp is gently affixed with little pads that are connected to the wires[1].

2.1.8.11 Genetic Tests

All forms of cancer, including brain tumors, alter the genetic material of the affected cells. Genes inherited from parents are not the same as these genetic defects. The defect is only in the structure of the tumor cells, not in the normal cells. Molecular genetics, often known as cytogenetics, is the study of these gene alterations. To find these gene alterations, a pathologist could perform specific tests on tumor cells. Physicians can use the results to tailor the treatment for that tumor[2].

2.1.9 Brain Tumor Further Tests and Investigations

After determining through the scan or scans that the patient has a tumor, the doctors may need to take a sample of the tumor for a pathologist (a medical professional who specializes in the causes, effects, and behavior of illnesses) to analyze. This process is referred to as biopsy. The pathologist will determine the nature of the tumor if it is malignant or non-malignant. Additional brain scans, such as the ones previously mentioned or an angiogram may also be necessary to assist the surgeon in organizing any surgery[1].

2.1.9.1 Biopsy

A biopsy involves taking a tiny sample of a tumor so a pathologist may examine it. Often, a preliminary diagnosis is made during the procedure; however, confirmation may take several days. The surgery is typically performed through a burr hole, which is a tiny incision the physician makes in the skull. A needle is inserted into the tumor via the burr hole. The needle is used to obtain a sample of the tumor. A customized frame affixed to the head and scanning equipment is often used to guide the needle to the tumor. This is known as a stereotactic biopsy. The patient will have a frame fitting prior to the biopsy. Thanks to recent developments, the technique can now sometimes be performed without the frame. This is referred to as image-guided surgery or frameless stereotaxis. Although they are occasionally performed under a local anesthetic, biopsies are typically performed under a general anesthesia[1].

Rather than offering medical care, the purpose of a biopsy is to aid in diagnosis. On the other hand, a biopsy may be used as a component of a patient's treatment in situations where a greater portion of the tumor is removed. The same risks that come with surgery also apply to biopsies. After the biopsy, there is a chance that the patient's symptoms will get worse or that it will result in seizures. The two types of biopsy procedures that are available are Open Biopsy and Closed Biopsy. Open biopsies are performed during craniotomies whereas a closed biopsy, also known as a stereotactic or needle biopsy, uses a needle to access and remove a small sample of tumor tissue from a hard-to-reach location[3].

2.1.9.2 Angiogram

An angiogram is an X-ray test that creates images of blood arteries. Sometimes, angiography is performed to show the blood supplies of a suspected brain tumor. Occasionally, the test results can assist physicians by indicating whether or not the tumor has spread and in determining the optimal course of treatment for a patient[1].

2.1.10 Brain Tumor Prognosis

The projected course of an illness is known as its prognosis. Although the precise course of the disease cannot be predicted, the prognosis may be impacted by a number of factors. These include age, family history, general health, kind of tumor, location of tumor, tumor grade, and genetic makeup; they also include how effectively the tumor reacts to treatment. Both high-grade and low-grade tumors have the potential to be fatal, however if the tumor is low grade or the surgeon is able to remove the entire tumor, the prognosis could be better. Certain tumors of the brain or spinal cord, especially gliomas, have the potential to recur and develop into a higher-grade tumor. In this situation, therapies including radiation therapy, chemotherapy, or surgery may be utilized to ease symptoms, preserve quality of life, and limit the tumor's growth for as long as feasible[2].

2.1.11 Brain Tumor Multidisciplinary Team

There are various therapeutic options for brain tumors, and each patient will require a distinct course of treatment. According to national guidelines published by the NICE (National Institute for Health and Care Excellence), each patient with a brain tumor should get care from a multidisciplinary team (MDT). Among them are Neuropathologists, Neuro-Oncologists, Neurologists, Neurosurgeons and Neuro-Radiologists. The following professionals are involved in cancer care coordination: clinical psychologists, neuro-oncology nurses, registered dieticians or nutritionists, medical oncologists, rehabilitation specialists, social workers, physiotherapists, occupational therapists, psychiatrists, exercise physiologists, speech therapists, and palliative care specialists. The patient's case will be discussed by the MDT, who will then put together a customized treatment plan for the patient. After their consultation, the patient will have a discussion with one of the MDT specialists regarding their treatment options. Brain tumors can pose a major risk to one's life. It is not always possible to cure patients, even with the best medicines available and the experience of the doctors involved in their care. In this instance, the goal of the care and treatment will be to control the symptoms that the tumor is producing[1].

2.1.12 Treating Brain Tumor

The goal of treating brain tumor may be to totally eliminate the tumor, limit its growth, or reduce swelling and decrease the tumor in order to relieve symptoms. Patients' age, medical history, overall health, and the kind, size, grade, location, genetic makeup of the tumor and the symptoms experienced will all influence the treatment plan that is recommended. Surgery might be the sole course of treatment required for a benign tumor whereas treatment options for a malignant brain or spinal cord tumor may include radiation therapy, chemotherapy, and surgery, which can be administered separately or in combination. To lessen symptoms, physicians may prescribe medications like steroids or anticonvulsants, or anti-seizure medications. Through a clinical trial, patients might get access to novel or altered treatments[2].

2.1.12.1 Steroids

The swelling around the tumor is decreased with the use of steroids. This lessens the amount of physical impairment the tumor may cause such as weakness or numbness and relieves headaches. Surgery becomes considerably simpler and safer as well. The most used steroid for treating brain tumors is called Dexamethasone. After a few weeks of treatment, side effects from steroids may manifest as increased hunger and weight gain, acne, muscle weakness, elevated blood sugar, disturbed sleep, and agitated and restless feelings. When the steroids are stopped, these adverse effects usually go away. The stomach lining may get irritated by steroid tablets, which may raise the risk of developing stomach ulcers. Anti-ulcer medications may be administered to the patient to lower the risk. An antacid medication may also be administered in tablet or liquid form[1].

2.1.12.2 Surgery (Craniotomy)

An operation to expose the brain by opening the head is called a craniotomy. A craniotomy is the creation of a hole (-tomy) in the skull (cranium). A neurosurgeon with expertise in brain and spine surgery performs the procedure. The goal of craniotomy is to remove the tumor completely while preserving the surrounding brain tissue. While some benign tumors may be able to be treated in this way, malignant tumors are almost never treated in this way because they infect adjacent brain tissue, which has to be left intact in order to prevent severe damage. When a neurosurgeon finds a malignant tumor, they usually remove as much of the tumor as they deem safe.

Typically, a general anesthetic is used throughout the procedure, so the patient won't feel anything and will remain unconscious the entire period. On the other hand, should the tumor be close to brain regions involved in critical functions, an "awake craniotomy" may be performed. This process enables the surgeon to monitor the patient’s brain activity during the surgery (e.g., by having him read aloud) in order to ensure that certain critical brain regions are not injured. The hair is shaved in a small horseshoe-shaped patch over the tumor's entry location. The brain and tumor underneath are revealed by making an incision (cut) in the scalp, peeling back a skin flap, drilling burr holes in the skull, and then cutting away a piece of bone (called a "bone flap").

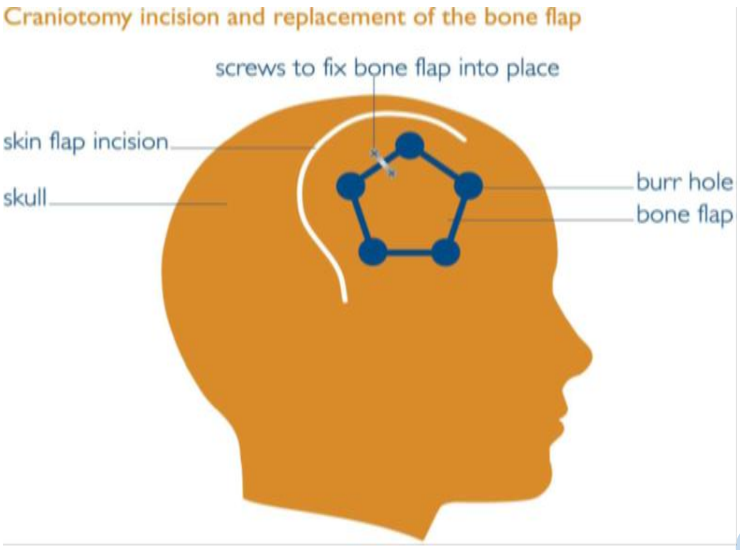


Figure 2.1: Craniotomy Incision and Replacement of the Bone Flap[1]

After that, the surgeon cut out the tumor as much as possible. This is referred to as "debulking or resection. Following the removal of some of the tumor cells, a pathologist will examine them to determine the type and grade of the tumor and provide recommendations for treatment. In addition to acting as a tumor treatment by removing as much of the tumor from the brain as is safe, the procedure also serves the dual purposes of obtaining a portion of the tumor for pathology examination (such as a biopsy). The scalp is sewn back together after the replacement of the bone flap. To stop movement and promote better healing, the bone flap is typically secured in place with tiny metal screws.

Following a craniotomy, people may have a variety of issues. The potential issues are contingent upon the specific region of the head that underwent surgery. Following the procedure, you may experience headaches, facial bruising and swelling, a stiff jaw that makes it difficult for you to open your mouth, and pain and discomfort while your head incision heals. The bone flap may also feel as though it is clicking or moving. It feels weird, yet this is not harmful. The flap of bone is not free. When it is replaced, it is fixed and will realign itself[1].

Other Types of Surgery

i. Endoscopic Transsphenoidal Surgery (Extraction of a Pituitary Tumor): The most frequently performed type of surgery for tumors close to the base of the brain, such as pituitary gland tumors, is endoscopic transsphenoidal surgery. An endoscope (a long, thin tube equipped with a light and camera) is inserted through the nose and into the base of the brain's skull by the surgeon in order to remove the tumor. An ear, nose and throat (ENT) surgeon may provide assistance for this kind of surgery. Additionally, a general anesthetic will be administered to the patient[2].

ii. Laminectomy (Removing a Spinal Cord Tumor): A laminectomy is the surgical procedure most frequently used to treat spinal cord tumors. During this surgery, the surgeon creates an incision through the skin, muscle, and vertebra in the spinal column to remove the tumor that is harming the spinal cord. A general anesthetic is typically used during a laminectomy.

iii. Computer Guidance: Nowadays, a computer navigation system is typically used to guide the surgeon during a craniotomy. This kind of operation is known as stereotactic surgery. Three-dimensional images of the brain and tumor are produced by the computer using the planning scan data. The surgeon can perform the procedure with extreme precision because the computer keeps an eye on the surgical equipment. Compared to non-computer-guided surgery, stereotactic surgery is safer, more precise, and necessitates a smaller incision in the skull.

Aftereffects of Surgery

i. Infection: There is a very small chance of getting an infection at the surgery site. Antibiotics are typically effective in treating this. A tiny percentage of patients might require surgery to remove the wound.

ii. Bleeding: This adverse reaction is uncommon but dangerous. The day following surgery, an MRI or CT scan will be performed to look for any swelling or bleeding.

iii. Swelling: Swelling might result from surgery. The pressure inside the skull (intracranial pressure) rises as a result of this swelling. The healthcare team will monitor and make an effort to lessen the edema.

iv. Other Adverse Reaction: The patient can still experience disorientation and vertigo, as well as weakness, difficulty speaking, and seizures. He might feel worse than he did before the operation, which might surprise him and his family or caregivers and make him fear that he isn't healing properly. These adverse effects are typical and frequently go better with time. Some people heal completely and are able to gradually resume their regular routines. In some situations, the location of the tumor or harm to nearby brain tissue may result in longer-term alterations to the speech, movement, and thought processes.

2.1.12.3 Radiotherapy

Radiation therapy, sometimes referred to as radiotherapy, is frequently used following surgery, mostly to treat malignant tumors. Radiation therapy is occasionally used to treat benign cancers as well. This can also be applied in cases when tumor cells are located in difficult-to-reach locations[3]. In radiotherapy, cancer cells in the treated area are killed or damaged by a carefully calibrated radiation dose. While side effects are conceivable, treatment is carefully designed to cause as little damage as possible to the healthy body tissue surrounding the tumor. Following surgery, radiation therapy is often used, sometimes in conjunction with chemotherapy. In order to determine the exact area that has to be treated, a radiation therapist will measure the body and do an MRI or CT scan before the radiation therapy begin[2].

In the event that the patient has undergone the surgery, radiotherapy treatment won't begin until after recovery. Gamma and X-rays are used in radiotherapy to either eliminate or harm the tumor. Radiotherapy helps to slow down the growth of the tumor and extend the amount of time before it regrows. Under certain conditions, radiation can eradicate the tumor entirely. The therapy is painless and entails lying on a specially made table for a few minutes. A customized mask will be fitted for the patient to wear over his face throughout the procedure. The mask causes discomfort for some people[1].

Radiation therapy is normally administered once a day, Monday through Friday, for several weeks. The frequency of treatments will vary depending on the size and kind of tumor. A shortened treatment course may be administered to certain individuals. The patient will lie on a table beneath a device known as a linear accelerator for the course of the treatment. Daily treatment is between ten to fifteen minutes. Although radiation therapy is painless, the patient will be informed of any potential adverse effects[2]. A radiotherapist administers radiotherapy in collaboration with an oncologist, a physician with specialized knowledge in cancer treatment. As part of the treatment, the oncologist and the neurosurgeon who conducted the operation will probably collaborate closely. They will address any issues or concerns the patient might have and provide him with an explanation of the radiation treatment. Although the hospital where the patient undergoes the operation may not always offer radiotherapy, there is always a tight relationship between the facilities handling the patient care. On rare occasions, radiotherapy may be administered by brachytherapy (by inserting implants into the tumor). This type of treatment is rare and is only suitable for a few people[1].

Options of Radiation Therapy

i. External Beam Fractionated Radiation: is the usual course of treatment administered in an outpatient clinic to all patients with high grade malignant gliomas[1].

ii. Stereotactic Radiosurgery: High radiation doses are directed at the tumor from a variety of angles during stereotactic radiosurgery. This type of radiation can be used to treat both benign and malignant cancers, but it works best on tumors with well-defined borders. It is frequently administered with the Gamma Knife® or the more recent CyberKnife® unit[1]. Stereotactic radiosurgery (SRS) is not a form of surgery; rather, it is a specialized form of radiation therapy in which no skull incisions are made. Some types of brain tumors are treated with it Radiation therapy is applied to the tumor with extreme precision using a specialist radiation machine. This indicates that a large amount of radiation is applied to the tumor and relatively little is applied to the healthy brain tissue around it. SRS is not appropriate for all brain tumors. It might be provided as an alternative to neurosurgery or in situations when neurosurgery is not feasible. It is frequently utilized for metastatic brain tumor that has spread to the brain from other part of the body. In addition, it is used infrequently to treat gliomas that have returned after previous therapy, as well as some cases of meningiomas, pituitary tumors, and schwannomas. Usually, only one to five SRS dosages are required. Depending on the type of radiosurgery used, a treatment session could last from 15 minutes to two hours. You will be required to wear a face mask or a frame during the procedure[2].

iii. Stereotactic Radiation Therapy (SRT): A lengthier course of radiation therapy can also be administered with a stereotactic radiosurgery machine, especially for benign brain tumors. This is referred to as stereotactic radiation therapy. The course of treatment consists of multiple small daily doses[2].

Side Effects of Radiotherapy

The most common side effects of radiation therapy include fatigue, short-term memory loss, skin inflammation that can cause temporary hair loss, and a temporary worsening of symptoms. Most patients experience hair growth in the first few months after treatment, but it may be patchy or thin and not as strong as before. Until a sufficient amount of hair has grown back, the head should be protected from direct sunlight by wearing a hat. Sometimes hair loss is irreversible. When their radiation treatment is coming to a conclusion, most patients experience fatigue. Individual differences in fatigue may be attributed to the specific brain region being addressed. Some individuals may simply feel slight fatigue and be able to carry out their daily routines. It is possible for some to suffer from extreme fatigue[1].

2.1.12.4 Radiosurgery

The two primary techniques used in radiosurgery are the modified linear accelerator (LINAC) and the gamma knife. Both techniques make use of a high-energy radiation dosage that may be targeted to an extremely specific location inside the brain. The LINAC treats a single location while sparing the surrounding tissue by using a single, high-energy radiation beam that arc around a single point. Each energy beam used by the gamma knife is too feeble to harm any healthy brain tissue in its path, but when combined, hundreds of them create a high-energy point at its center.

In contrast to radiation therapy, which may take numerous sessions spread out over weeks or months, radiosurgery is usually finished in a single session and does not necessitate an overnight stay. Following treatment, there won't be any side effects from radiation or craniotomy, allowing the patient gets back to your regular schedule right away. Deep-rooted brain tumors that may be challenging to access without harming nearby healthy brain tissue are believed to be better candidates for radiosurgery than conventional forms of treatment. Gliomas are not typically treated with radiosurgery.

2.1.12.5 Chemotherapy

Chemotherapy is the application of medication to kill tumor cells in a manner akin to how antibiotics are used to kill bacteria[1]. It is recommended when surgery is insufficient to eliminate a tumor, which is typically the case with higher-grade tumors[3]. The medications prevent cancer cells from proliferating and spreading by injuring them. Chemotherapy is typically advised when a cancer has spread or poses a risk of doing so. Different kinds of chemotherapy exist. Certain medications are administered orally, while others are infused intravenously. Following surgery, patience could receive radiation therapy along with chemotherapy. This is referred to as chemoradiation[2]. An oncologist with training in both radiation and chemotherapy is typically the one to administer chemotherapy.

Chemotherapy side effects might differ significantly based on the specific medication being taken. It is possible for side effects to include: nausea or vomiting, exhaustion, lack of energy, increased risk of infection, appetite loss, mouth sores and ulcers, diarrhoea or constipation, skin rash, breathlessness from low platelet or red blood cell levels, increased risk of abnormal bleeding, damage to the testicles or ovaries, which can result in infertility, and decreased bone marrow production of blood cells. Regular blood tests may be necessary to monitor any side effects. Chemotherapy treatments for brain and spinal cord cancers seldom cause complete hair loss, but they can cause a person hair to thin or become spotty in certain cases[2]. Some of the most recent chemotherapy medications, including temozolomide, have fewer adverse effects and are frequently well tolerated. They are accessible as pills, so patients don't have to stay in the hospital to take them[1].

There are three forms of chemotherapy:

i. Chemotherapy Wafers: During surgery, carmustine medication, also known as

BCNU, are placed straight into a high-grade glioma. To destroy tumor cells, the Gliadel® wafer gradually dissolves over two to three weeks.

ii. Intravenous Chemotherapy: This is when chemotherapy is administered intravenously i.e., through a vein. Among the high-grade gliomas are: BCNU, Nitrosurea Vinca alkaloids: platinum and vincristine Comparables: cisplatin and carboplatin.

iii. Oral Chemotherapy: Chemotherapy administered orally is known as oral chemotherapy. Temodar©, procarbazine (Matulane®), lomustine (CCNU), and TMZ or temozolomide are a few examples. For high grade gliomas, TMZ with radiation is the usual course of treatment. Anaplastic oligidendrogliomas are treated by certain physicians using procarbazine, vincristine, and CCNU (often referred to as PVC chemotherapy). For brain malignancies, oral chemotherapy is not always beneficial. This is because of the body's natural built-in defense mechanism in the brain and cerebro-spinal fluid. The blood-brain barrier, which keeps dangerous chemicals out of the central nervous system, is this defense mechanism[3].

2.1.12.6 Proton Beam Therapy

Proton beam therapy damages and kills tumor cells by using charged particle, also called beams of protons. Proton beams stop when they come into contact with the tumor, unlike the X-rays used in traditional radiation. This indicates that less of the surrounding healthy tissue has been harmed by the tumor[1]. When cancer is close to delicate structures like the brain, eyes, or spinal cord, proton therapy can be helpful[2].

2.1.12.7 Other Treatments

i. Anti-epilepsy Drugs: Seizures (epileptic fits) are managed with anti-epilepsy drugs (AEDs). AEDs come in wide varieties. The specific circumstances will determine which AED is prescribed for the patient.

ii. Pain-Relief Drugs: The primary purpose of painkillers is to manage headaches. To reduce nausea and vomiting, they may be taken in combination with other medications. There are numerous kinds of painkillers, and the doctors prescribe the one that will work best for a patient.

iii. Complementary Treatments: Complementary therapies like aromatherapy or reflexology are often given to brain tumor patients. There are also other food and vitamin supplements to choose from. The majority of medical professionals would only advise utilizing these in addition to accepted therapies, never as a replacement for them. Although complementary therapies and treatments are unlikely to specifically heal brain tumor, they may helpful in improving the overall wellbeing.

2.1.12.8 New Treatments for Brain Tumor

Currently under development are novel treatments such as immunotherapy, photodynamic therapy, gene therapy, and anti-angiogenesis therapies.

i. Photodynamic Therapy (PDT): The development of photodynamic therapy (PDT) is intended to treat brain tumors that cannot be surgically removed and to supplement surgery. The patient receives a medication that causes the tissue to become light-sensitive, typically by intravenous injection but occasionally straight into the tumor. The tumor is then targeted by a laser beam. This triggers the drug's light-sensitive ingredient, which destroys the tumor cells.

ii. Gene Therapy: Gene therapy entails introducing genes (DNA) into the tumor in order to either replace damaged genes or control the tumor's growth or to introduce particular genes that can stop the tumor from developing and possibly cause the tumor cells to die.

iii. Immunotherapy Drugs: The goal of immunotherapy medications is to improve the body's immune system's capacity to recognize and eliminate cancerous cells.

iv. Anti-Angiogenesis Drugs: Anti-angiogenesis medications stop new blood vessels from growing inside and around the tumor, preventing an increase in blood flow to the area. As a result, tumor growth slows down and tumor cells die[1].

v. Immunostimulatory Molecules: A monoclonal antibody, or lab-made antibody used for targeted therapy, called imipomumab (YervoyTM) is being evaluated for glioblastoma after showing promise in treating melanoma. It helps eliminate undesired tumor cells by stimulating the immune system.

vi. Clinical Trials: Studies called clinical trials are made to put the most promising new medicines to the test on actual patients. A person may take part in a clinical trial to attempt a novel and effective treatment approach, to help develop future treatments, or to assist in the search for a cure. Patients must meet specific medical requirements in order to participate in most clinical trials.

vii. Targeted Therapy: Targeted therapies use particular components of a cell, such as chemicals or processes necessary for the proliferation of cancer cells (i.e., cell proteins), as a point of focus. A targeted therapy can interrupt specific cell processes by attaching itself to a specific protein in a cancerous cell. As an illustration, Bevacizumab (Avastin©), an FDA-approved targeted medication interferes with a tumor's capacity to produce new blood vessels.

viii. Electric Field Treatments: A novel technique called electric field treatments uses a device called NovoTTFTM (by Novicure) that is applied along the scalp to kill brain tumor cells. It offers tiny electric currents (electrodes) that might halt tumor cell growth while sparing healthy brain tissue.

ix. Vaccine Therapy: A promising treatment that is still being developed is vaccine therapy. The immune system of the patient is used to identify and combat cancer cells. The body's natural defenses against cancer can be strengthened, directed, or restored by substances derived from brain tissue or created in a lab, much like how a flu vaccine helps the body fight the flu[3].

x. Palliative Treatment: Palliative care addresses cancer symptoms without attempting to treat the underlying cause of the illness, thereby enhancing quality of life. Palliative treatment can be used at any stage of advanced cancer. Moreover, it inhibits the growth of cancer. Palliative care can assist with various symptoms as well as pain relief. Surgery, radiation therapy, chemotherapy, and other medications or supplements are possible forms of treatment. One component of palliative care is palliative treatment, when a group of medical specialists works to address the social, cultural, emotional, physical, and spiritual needs of patient. Services for palliative care can be obtained in a hospital, residential care facility, or at home[2].

2.1.13 Utilizing Deep Learning Algorithms in the Detection and Classification of Brain Tumor

Deep learning (DL) architecture is one of the amazing attempts to use AI to identify brain tumors from MRI images[7]. DL is also used for medical imaging and a variety of Computer Aided Diagnosis (CAD) systems, demonstrating the enormous potential of these approaches in automating diagnosis. Convolutional neural networks, on the other hand, are among the other approaches that have continued to operate well in recent decades for classifying radiological images[8]. Deep learning algorithms, such as these kinds of methods, are used in many different applications, one of which is pattern recognition[9].

Conventional machine learning techniques approach issues by decomposing them into smaller components, including mathematical modeling, data collection and cleansing, feature extraction and selection, training, optimization, and model assessment. These kinds of approaches require a long modeling process, which adds to the computing cost and simulation time for large-scale applications. Unlike these approaches, deep neural networks (DNNs) are end-to-end learning techniques that can reduce the cost of creating and extracting complex features by combining multiple processing steps into a single DNN. All that DNN needs to achieve competitive and accurate performance is enough data and the right adjustments to its hyper-parameters[10].

Deep learning (DL) based on convolution neural networks (CNNs) has made imaging-based artificial intelligence (AI) solutions possible. The goal of DL-guided solutions is to support clinical judgment[104]. CNN is the state-of-the-art and latest development in the field of machine learning, which is used in the diagnosis of diseases based on medical imaging, especially CT and MRI scans. Since CNN does not require preprocessing or feature extraction prior to the training procedure, it has become increasingly popular in the classification and grading of medical imaging[11]. CNNs are typically used to handle raw images and are typically meant to reduce or eliminate certain data pre-processing stages. In the following order, the layers that make up CNN are: input, convolution, RELU, fully connected, classification and output layers. Convolution, which is carried out using trainable filters with predefined specifications that are modified throughout the training phase, and down sampling make up the majority of the CNN process[12].

Generally, there are two primary kinds of deep learning applications for brain tumor classification: the first classifies brain MRI images into normal and abnormal, and the second divides aberrant brain MRI images into multiple types of brain cancer. The CNN has been utilized for brain tumor detection and grading, and researchers are paying attention to it as a potent tool in the field of disease detection and classification. This will enhance the accuracy of the brain tumor grading that has been detected, assist doctors in formulating the ideal treatment plan, and ultimately increase the healing percentage[13]. Researchers have shown the immense potential of imaging techniques to lessen the burden that medical experts bear[14]. It also makes it possible to provide more assistance with patient care, lessen burnout, and lower patients' overall medical expenses.

2.1.14 Machine Learning

Every aspect of humans' life is digitally recorded, and we live in the age of data, where everything is linked to a data source[15, 16]. In the modern electronic world, for example, there is an abundance of diverse types of data, including data from the Internet of Things (IoT), cybersecurity, smart city, business, smartphone, social media, health, COVID-19, and many more sources. There is a rising amount of unstructured, semi-structured, and structured data available every day. By drawing conclusions from these data, numerous intelligent applications in the pertinent fields can be constructed. For example, relevant cyber security data can be utilized to build an automated and intelligent cybersecurity system driven by data; similarly, relevant mobile data can be used to create personalized, context-aware, smart mobile applications, and so on[16]. Therefore, the real-world applications that rely on data management technologies and methodologies capable of intelligently and quickly extracting insights or meaningful knowledge from data are desperately needed.

In the context of data analysis and computers, artificial intelligence (AI) and machine learning (ML) have advanced significantly in recent years, allowing programs to generally operate intelligently[17]. Artificial intelligence (AI) has a branch called machine learning (ML) that allows computers to learn from data without explicit programming[18]. Machine learning depends on input, such as training data or domain knowledge, to understand objects, domains, and the connections between them, much like the human brain does. It is necessary to define entities before starting deep learning. The foundation of machine learning is observation and data, such as case studies, first-hand knowledge, or guidelines. It searches the data for patterns so that it can then draw inferences from the provided instances[19].

Intelligent systems that can identify patterns in data, anticipate outcomes, and pick up new knowledge can be developed using machine learning[18]. The main goal of machine learning is to create systems that can learn from their experiences on their own, without human help, and adapt their behavior accordingly[20]. The creation of more reliable and accurate models that can evaluate intricate information and offer insights to guide decision-making is one benefit of applying ML techniques[18]. The idea of machine learning has been around for a while. Arthur Samuel, an IBM computer scientist who pioneered artificial intelligence and computer games, is credited with coining the term "machine learning". Samuel created checkers-playing computer software. The more often it was used, the more experience the program gained and the more algorithms it could anticipate. Machine learning is the science of creating algorithms that can anticipate and learn from data[19].

2.1.14.1 Machine Learning Algorithm

The key to developing intelligent data analysis and associated intelligent and automated real-world applications is understanding artificial intelligence (AI), and specifically machine learning algorithms (ML). Four main types of machine learning algorithms can be distinguished: semi-supervised, supervised, unsupervised, and reinforcement learning[21]. To effectively create data-driven systems, machine learning algorithms such as association rule learning, regression, data clustering, feature engineering and dimensionality reduction, and classification analysis are available[22, 23].

Additionally, deep learning is derived from artificial neural networks, which belong to a larger family of machine learning techniques and are capable of doing intelligent data analysis[24]. It is therefore difficult to choose an appropriate learning algorithm that fits the intended application in a given domain. The rationale is that various learning algorithms have distinct goals, and even within a same category, the results of various learning algorithms can differ based on the properties of the data[25].

Understanding the fundamentals of different machine learning algorithms and how they apply to different real-world domains, including smart cities, cybersecurity, IoT, business, recommendation, healthcare, COVID-19, context-aware systems, sustainable agriculture, and many more, is crucial[26].

2.1.14.2 Real World Data

Usually, machine learning algorithms take in and analyze data to discover relevant patterns regarding people, transactions, events, business procedures, and so forth. Generally speaking, data accessibility is seen as essential to building data-driven real-world systems or machine learning models[16]. There are many different types of data, including unstructured, semi-structured, and structured data[22, 27]. Additionally, another form that often represents data about the data is called metadata.

i. Structured Data: It is utilized by an entity or computer program, has a clearly defined structure, and adheres to a data model with a regular order. It is also highly organized and accessible. Stored in tabular format, structured data are usually kept in well-defined schemes like relational databases. Structured data includes things like names, dates, addresses, credit card numbers, stock information, geolocation, and so on.

ii. Unstructured Data: Unstructured data, has a predetermined format or organization, making it considerably more challenging to collect, handle, and analyze—mostly consisting of text and multimedia content. Unstructured data includes, but is not limited to, sensor data, emails, blog posts, wikis, word processing documents, PDF files, audio files, videos, images, presentations, web pages, and many more kinds of business documents.

iii. Semi-Structured Data: Although semi-structured data is not kept in a relational database like the previously discussed structured data, it does have some organizational characteristics that facilitate analysis. Semi-structured data includes things like HTML, XML, JSON documents, NoSQL databases, and more.

iv. Metadata: It is data about data, not the typical form of data. Data are just the materials that can be used to classify, quantify, or even document something in relation to an organization's data attributes. This is the main distinction between data and metadata. However, metadata provides a description of the pertinent data, making it more meaningful to data users. Document authors, file size, date of creation, keywords used to characterize the content, etc. are a few fundamental examples of document metadata.

Researchers in the fields of data science and machine learning employ a variety of popular datasets for a variety of objectives. These include, for instance, datasets related to cyber security, smartphones, the Internet of Things, agriculture, e-commerce, health, and many other application domains. The data may be in any of the several forms mentioned above, depending on the specific use case in the real world. In order to examine this kind of data inside a specific problem domain and derive insights or practical knowledge for developing intelligent applications for the actual world, many machine learning approaches can be applied based on their capacity to learn[26].

2.1.14.3 Machine Learning Techniques

Supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning are the four basic categories into which machine learning algorithms fall[21].

i. Supervised Learning: In supervised learning, students use labeled examples to apply past knowledge to new data in order to forecast future events. The learning approach creates an inferred function to forecast output values by examining a known training dataset. Once the system has received enough training, it may offer targets for any new input. It can also identify flaws in the model and make necessary adjustments by comparing its output to what is proper and planned[19]. Developing a model that can precisely forecast the outcome for novel, unknown inputs is the aim of supervised learning[18].

   When certain objectives are determined to be achieved from a given set of inputs, supervised learning, which is a task-driven technique, is implemented[16]. The two most popular supervised tasks are regression, which fits the data, and classification, which divides the data. One example of supervised learning is text classification, which is the process of predicting the class label or sentiment of a text, such as a tweet or a product review. Applications including audio and image identification, natural language processing, and prediction tasks frequently employ it. Neural networks, decision trees, logistic regression, and linear regression are a few types of supervised learning algorithms[18].

ii. Unsupervised Learning: When a computer is trained on unlabeled data, it must discover underlying patterns or structure on its own without human assistance. This process is known as unsupervised learning. Through the use of dimensionality reduction or clustering algorithms, the algorithm gains the ability to recognize hidden relationships and groupings within the data[18]. Accuracy of the system's output is never guaranteed. Rather, it extrapolates datasets expected result[19]. Unsupervised learning aims to extract novel knowledge and insights from the data without imposing any preconceptions about the desired outcome[18].

Unsupervised learning is frequently used for exploratory reasons, groupings in findings, generative feature extraction, and the identification of significant patterns and structures. Clustering, density estimation, feature learning, dimensionality reduction, identifying association rules, anomaly detection, and so on are among the most popular unsupervised learning tasks[26]. Applications like anomaly detection, recommendation engines, and data visualization frequently employ it. Principal component analysis (PCA), autoencoders, k-means clustering, and hierarchical clustering are a few instances of unsupervised learning methods[18].

iii. Semi-Supervised Learning: Because semi-supervised learning uses both labeled and unlabeled data, it can be thought of as a hybridization of the supervised and unsupervised approaches[22, 16]. It therefore lies in the middle of "without supervision" and "with supervision" learning. In real-world scenarios, when unlabeled data are abundant and labeled data may be scarce in certain settings, semi-supervised learning can be beneficial. A semi-supervised learning model's ultimate objective is to produce a better prediction result than one might obtain from the model utilizing just the labeled data. Semi-supervised learning finds application in machine translation, fraud detection, data labeling, and text categorization, among other areas[21].

iv. Reinforcement Learning: Reinforcement learning algorithms interact with their surroundings by taking actions and assessing the results. The two most important components of reinforcement learning are delayed rewards and trial-and-error learning[19]. Reinforcement learning involves the computer learning by making mistakes and getting feedback for those mistakes in the form of incentives or punishments. Through exploration and experience, the algorithm learns to make a series of actions that maximize a cumulative reward over time[18]. The goal of reinforcement learning is to increase efficiency by enabling software agents and computers to automatically determine the best course of action in a given situation. The reinforcement signal, which is a straightforward reward feedback, is what the agent needs to decide which behavior is preferred[19]. It is an effective technique for developing AI models that can help improve the operational efficiency or automate more complex systems like robots, autonomous driving, recommendation engines, manufacturing, and supply chain management. It is not, however, the best method for resolving simple or elementary issues[26]. Algorithms for reinforcement learning include actor-critic approaches, policy gradient methods, and Q-learning[18].

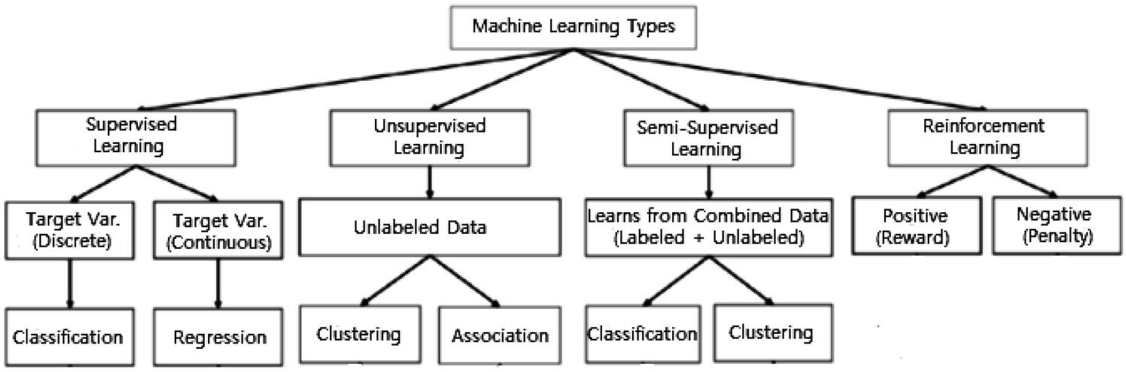


Figure 2.2: Types of Machine Learning Techniques[17]

Therefore, depending on the type of data and the desired result, different types of machine learning approaches can be very important in building effective models in different application areas based on their capacity for learning.

2.1.15 Applications of Machine Learning

i. Image Recognition: Image recognition is one of the most prominent uses of machine learning. It is a technique for classifying and identifying an object or feature in a digital image. Further analysis methods including pattern recognition, face detection, and face recognition also employ this technique.

ii. Voice Recognition: ML software can measure spoken words by taking a set of integers that correspond to the speech signal. Google Maps, Apple's Siri, and Amazon's Alexa are a few well-known apps that use speech recognition.

iii. Predicting Road Traffic Patterns: using Google Maps as an example, the software gathers a lot of information about current traffic patterns when location is entered on the map, this information is used to forecast future traffic patterns and determine the quickest path to our destination.

iv. E-commerce Product Recommendations: Product suggestion is a key component of almost any e-commerce website and requires the advanced application of machine learning algorithms. Websites monitor user behavior based on past purchases, surfing preferences, and cart history before employing AI and machine learning to make product recommendations.

v. Self-Driving Vehicle: An unsupervised learning system, mainly reliant on machine learning techniques, powers self-driving cars. The car can get data from cameras and other sensors about its environment, comprehend it, and decide what to do with the help of this algorithm.

vi. Catching Junk mail: Email service providers, create programs with spam filters that identify incoming emails as spam and move them to the spam folder using a machine learning algorithm.

vii. Catching Malware: There are two main steps in the machine learning (ML) process for detecting malware. Firstly, in order to create an appropriate collection of features, it analyzes suspicious activity in an Android environment. Next, train the system to apply machine learning and deep learning techniques on the generated features in order to detect future cyber-attacks in similar environments.

viii. Virtual Personal Assistant: Virtual personal assistants offer access to pertinent information through text or voice. When a user submits a question into the system, the personal assistant finds the answer by searching for it or by remembering previous queries the user has asked. Natural language processing, text-to-speech conversion, speech recognition, and speech-to-text conversion are a few common machine learning (ML) techniques used in virtual assistants.

ix. Online Scam Recognition: Fraud detection is one of the most important uses of machine learning. The machine learning algorithm thoroughly reviews each customer's profile after they finish a purchase, looking for any odd trends that could indicate online fraud.

x. Stock Market and Day Trading: In the context of day trading and the stock market, machine learning uses algorithmic trading to extract relevant data in order to support or automate critical investment activities. ML is used to successfully manage portfolios and make stock buying and selling decisions[28].

2.1.16 Challenges and Research Directions in Machine Learning

Generally speaking, the type and qualities of the data as well as the functionality of the learning algorithms determine how successful and efficient a machine learning-based solution is. It is not easy to gather data in the important domains, like cyber security, IoT, healthcare, and agriculture, even though the contemporary cyberspace makes it possible to produce a large amount of data very frequently. For this reason, gathering pertinent data for the intended machine learning-based applications—such as those for smart cities—and managing it are crucial to more research. Therefore, in working with the real-world data, a more thorough examination of data collection techniques is required[26].

Furthermore, a large number of missing values, ambiguous values, outliers, and meaningless data may be present in the historical data. The availability and quality of training data are greatly impacted by machine learning techniques, which also have an impact on the final model. It is therefore a difficult effort to precisely clean and pre-process the various data collected from diverse sources. Thus, to effectively employ the learning algorithms in the related application domain, one must either propose new data preparation approaches or alter or improve existing pre-processing methods[26].

Many machine learning techniques exist to examine the data and derive insights. It is therefore difficult to choose an appropriate learning algorithm that is appropriate for the intended application. The rationale is that the results of various learning algorithms can differ based on the properties of the data[25]. Choosing the incorrect learning algorithm could lead to unforeseen results that could reduce the efficacy and accuracy of the model and cause effort loss. When it comes to creating models, machine learning techniques can be applied directly to resolve a wide range of real-world problems in various fields, including cybersecurity, smart cities, and healthcare. On the other hand, future research in this field may focus on the hybrid learning model, which includes the ensemble of methods, improving or changing the current learning approaches, or creating new learning methods.

As a result, both the data and the learning algorithms are crucial to the final success of a machine learning-based solution and its related applications. Machine learning models may become ineffective or yield poorer accuracy if the data are unsuitable for learning, such as non-representative, low-quality, irrelevant features, or insufficient quantity for training. Thus, for a machine learning-based solution and ultimately the development of intelligent applications, efficiently handling the data and a variety of learning algorithms is crucial[26].

2.1.17 Deep Learning

A branch of machine learning called Deep Learning, studies algorithms that draw inspiration from the composition and operations of the human brain. Deep learning requires no human-designed procedures to function, DL maps the provided input to particular labels by utilizing a vast quantity of data. Artificial neural networks, or ANNs, are used in the construction of deep learning (DL), with multiple layers of algorithms that each provides a unique interpretation of the data that is supplied to them[30, 31].

Pre-processing, feature extraction, careful feature selection, learning, and classification are the sequential processes needed to complete the classification assignment using traditional machine learning approaches. Moreover, a major influence on the effectiveness of machine learning approaches is feature selection. Inaccurate class distinction may result from biased feature selection. On the other hand, unlike traditional Machine Learning techniques, DL can automate the learning of feature sets for several tasks[30, 32]. Learning and categorization can be completed simultaneously with the use of DL. On tasks like image categorization, deep learning performance has recently surpassed human performance[33].

A neural network uses a lot of data to learn from in an effort to simulate the functioning of the human brain. Many Artificial Intelligence applications that enhance the way tools and systems deliver services—like voice-activated technology and credit card fraud detection—are powered by deep learning. Deep learning clusters data to make precise predictions in an effort to resemble the human brain, whereas machine learning employs algorithm-driven data reprocessing. Deep learning algorithms carry out tasks iteratively, fine-tuning them each time to yield better results. The learning of the algorithms is driven by massive volumes of data[34].

Deep learning is made possible by the astounding amount of data produced every day[35]. The main factor propelling the development of deep learning skills is the enormous increase in data generation. Virtual assistants, self-driving cars, chatbots, facial recognition, speech recognition, and medical science are just a few of the applications that deep learning can serve. Organizations and businesses may automate operations and perform things more effectively, quickly, and affordably with the use of deep learning[34]. The usage of deep neural networks, which comprise numerous layers of interconnected nodes, is a fundamental aspect of deep learning. By identifying hierarchical patterns and features in the data, these networks are able to learn complicated representations of the data. Without the need for human feature engineering, deep learning algorithms can automatically learn from data and get better at analyzing data[36].

Research on traditional multilayer perceptron (MLPs) led to the development of DL. While traditional MLPs have only one hidden layer, DL designs frequently incorporate multiple hidden layers, making it possible to train models using either labeled or unlabeled raw data sets. The DL model learns ever-deeper, higher-level characteristics as it processes the raw or labeled data as it enters and moves through each hidden layer. At the end, high-dimensional discriminant feature vectors are created by encoding the raw or labeled data[37].

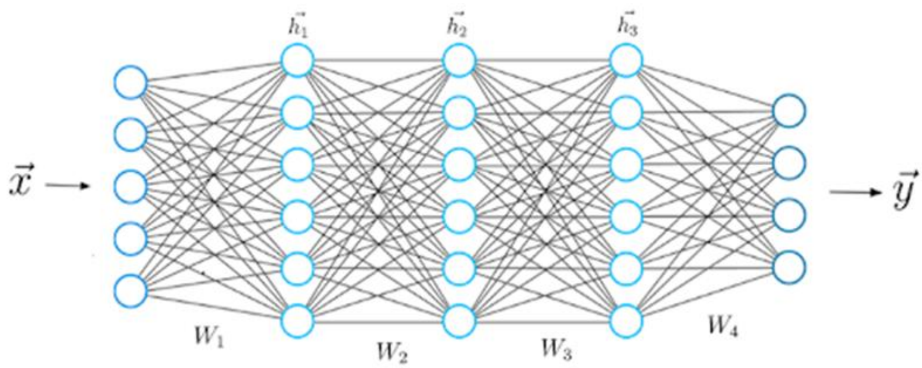


Figure 2.3: Neural Network Architecture[38]

While many Deep Learning architectures have been proposed, almost all of them possess convolutional, pooling, fully connected, and softmax layers as common elements[39]. DL models build on the fundamental architecture of the MLP by adding multiple hidden layers to learn complex regressions and map input data to target outputs. Due to their strong regression capabilities, DL models have many, sometimes millions, of parameters that need to be adjusted. Because supervised learning is the primary method of training DL models, building DL models necessitates a significant investment of time and training data sets. In the early days of DL, it was challenging to develop DL models. Nonetheless, two approaches have assisted researchers in getting over the previously described challenges. First, researchers can now easily and effectively develop DL models with the help of recent advances in computing hardware and software[37].

For specific kinds of DL models, different DL platforms, such as central processing units, graphics processing units (GPUs), and tensor processing units, have unique benefits. Secondly, the majority of remote sensing applications entail big data analysis, and training DL models is a straightforward process using large-scale annotated remote sensing data sets[40]. Deep learning models are also known as deep neural networks since most deep learning techniques make use of neural network topologies. Typically, "deep" refers to the quantity of hidden layers in a neural network. Deep networks include up to 150 hidden layers, compared to only 2-3 in traditional neural networks. Neural network designs that learn features directly from the input, eliminating the requirement for human feature extraction, and big quantities of labeled data are used to train deep learning models[41].

2.1.18 Classification of DL Approaches

Reinforcement learning, as well as supervised and unsupervised machine learning, can be accomplished with deep learning.

i. Deep Supervised Learning: The deep learning method known as "deep supervised learning" is when a neural network is trained to predict or categorize data using labeled datasets. Here, we input the target variables in addition to the two input features. Backpropagation is the process by which a neural network learns to generate predictions based on the cost or mistake resulting from the discrepancy between the projected and real goal. Convolutional neural networks (CNNs), recurrent neural networks (RNNs), and deep neural networks (DNNs) are examples of deep learning algorithms that are used for a variety of supervised tasks, such as sentiment analysis, language translation, image categorization and recognition[35]. Long short-term memory (LSTM) techniques and gated recurrent units (GRUs) are also included in the RNN category. This technique's primary benefit is its capacity to gather data or produce data output based on past information. The drawback of this method is that if the training set lacks examples that belong in a class, the decision boundary may be overstretched. In general, this method of learning with excellent performance is easier to use than other methods.

ii. Deep Semi-Supervised Learning: This method uses semi-labeled datasets as the basis for its learning process. This method is sometimes combined with deep reinforcement learning (DRL) and generative adversarial networks (GANs). Furthermore, RNNs, which comprise LSTMs and GRUs, are also used in partially supervised learning. Reducing the quantity of labeled data required is one benefit of this method. However, one drawback of this method is that it may result in inaccurate conclusions due to irrelevant input features in the training data. One of the most often used applications of semi-supervised learning is the text document classifier. Text document classification tasks benefit greatly from semi-supervised learning since large-scale tagged text document acquisition is challenging.

iii. Deep Unsupervised Learning: The deep learning method known as "deep unsupervised learning" involves using unlabeled datasets to teach a neural network how to find patterns or cluster data. Target variables are absent in this case. The hidden linkages or patterns in the datasets must be discovered by the machine on its own[36]. For unsupervised tasks including clustering, dimensionality reduction, and anomaly detection, deep learning methods such as generative adversarial networks (GANs), auto-encoders, and limited Boltzmann machines are employed. Furthermore, a variety of RNN applications, such as GRUs and LSTM techniques, have been used for unsupervised learning. The primary drawbacks of unsupervised learning are its processing complexity and inability to deliver precise information about data sorting[33].

iv. Deep Reinforcement Learning: Deep reinforcement learning is a type of deep learning where an agent gains the ability to maximize a reward signal by making decisions in its surroundings. By acting and observing the rewards that follow, the agent engages with the environment. Policies or a collection of activities that optimize the cumulative reward over time can be learned by deep learning. Tasks like robotics and gaming are reinforced by deep reinforcement learning algorithms such as Deep Q networks and Deep Deterministic Policy Gradient (DDPG) [36].

Supervised learning works with given sample data, whereas reinforcement learning works with interactions with the environment. In 2013, Google Deep Mind was used to build this approach[42]. Afterwards, other improved methods reliant on reinforcement learning were developed. Reinforcement learning's primary flaw is that its learning rate can be affected by several factors. Reinforcement learning takes a lot of time and computation, especially when the workspace is large[33].

2.1.19 Types of Deep Learning Architectures

In deep learning, feedforward neural networks, recursive neural networks (RvNNs), recurrent neural networks (RNNs), and convolutional neural networks (CNNs) are the architectures that are most frequently utilized. There are advantages to each of these architectures for particular use situations. They all work, though, pretty much the same way by feeding in the data, and the model determines by itself whether or not it has correctly interpreted or decided on a particular data point.

2.1.19.1 Feedforward Neural Networks (FNNs)

The most basic kind of artificial neural network (ANN) is a Feedforward Neural Network, which has a linear information flow. For applications like speech recognition, image classification, and natural language processing, FNNs have been frequently used[36].

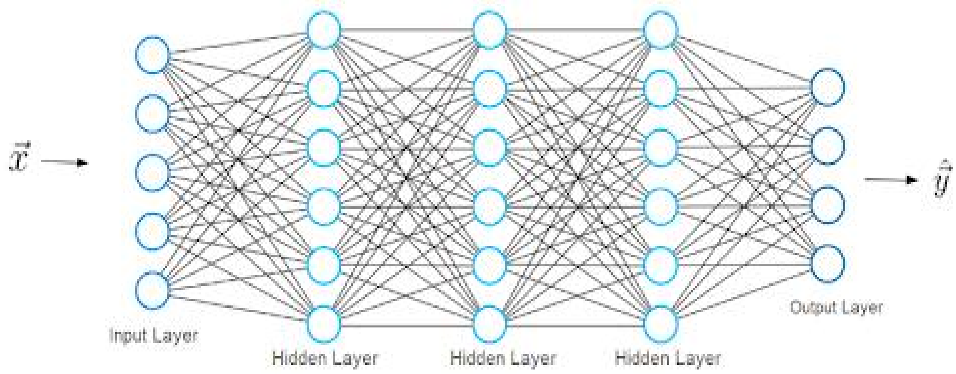


Figure 2.4: A Typical Feedforward Neural Network[38]

2.1.19.2 Recursive Neural Networks (RvNNs)

Recursive neural networks are able to classify the outputs using compositional vectors and make predictions in a hierarchical framework. The main source of inspiration for the development of RvNN is recursive auto-associative memory (RAAM). For processing objects with randomly formed structures, such as trees or graphs, the RvNN architecture is constructed. This method takes a variable-size recursive data structure and turns it into a fixed-width distributed representation. An introduced back-propagation through structure (BTS) learning system is used to train the network. The BTS system may accommodate a treelike structure and uses the same methodology as the general-back propagation algorithm. The network is trained to reproduce the input-layer pattern at the output layer by auto-association. RvNN performs exceptionally well in the context of NLP[33].

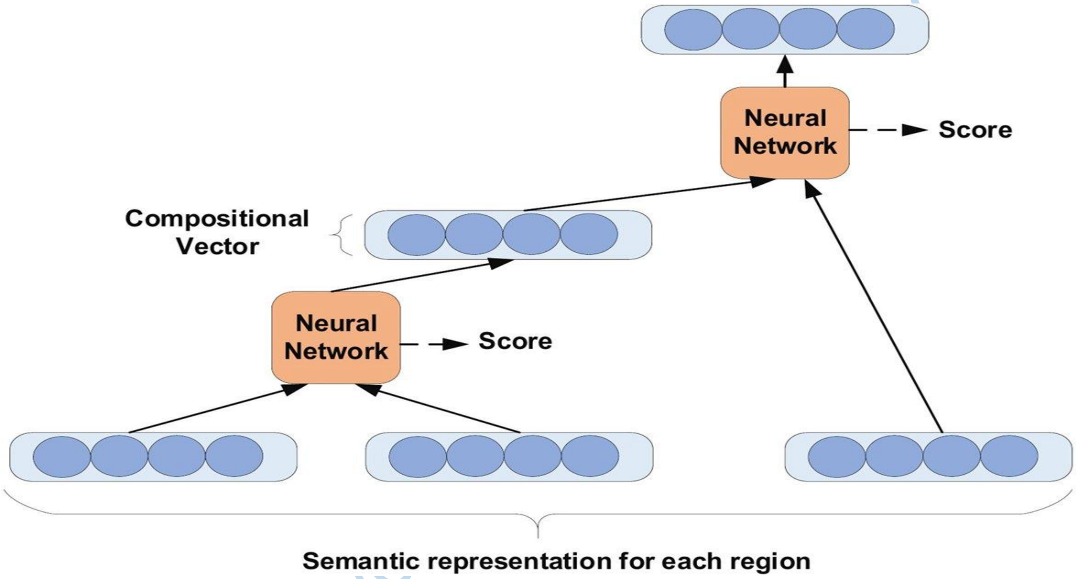


Figure 2.5: RvNN Tree[33]

2.1.19.3 Recurrent Neural Networks (RNNs)

One kind of neural network called recurrent neural networks (RNNs) is capable of processing sequential data, including time series and plain language. Due to their ability to retain an internal state that records details about the preceding inputs, RNNs are highly suitable for tasks including language translation, natural language processing, and speech recognition[36].

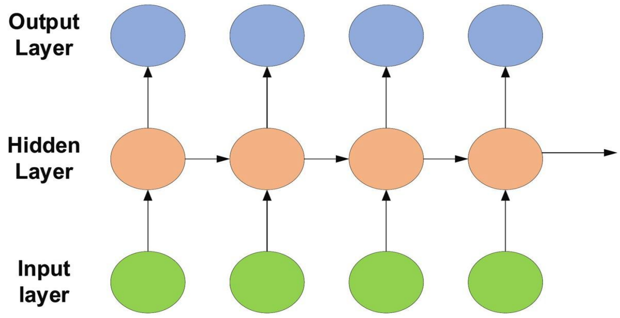


Figure 2.6: Recurrent Neural Networks Architecture[33]

2.1.19.4 Convolutional Neural Networks (CNNs)

Convolutional neural networks are incredibly powerful deep learning techniques. They have been applied to yield innovative results for computer vision applications[20]. CNNs are ideal for applications like object detection, image segmentation, and image classification because they can automatically extract pertinent information from the images without human oversight[36]. In general, CNNs are made to handle data in numerous arrays, and they work particularly well for this purpose. A softmax layer, convolutional, pooling, and fully linked layers are all present in a typical CNN architecture. In order to extract or improve spatial information from the input data, convolutional layers are utilized.

To add nonlinearity to the model, an activation function is added to the convolution-layer output. By lowering the spatial dimensions of the input data, pooling layers allow the following layers to focus on a wider variety of input patterns by lowering the number of trainable parameters in those levels. The amount of appropriate high-level features that are extracted rises when the input is processed at deeper layers. Lastly, for data classification, the fully connected and softmax layers are employed[37].

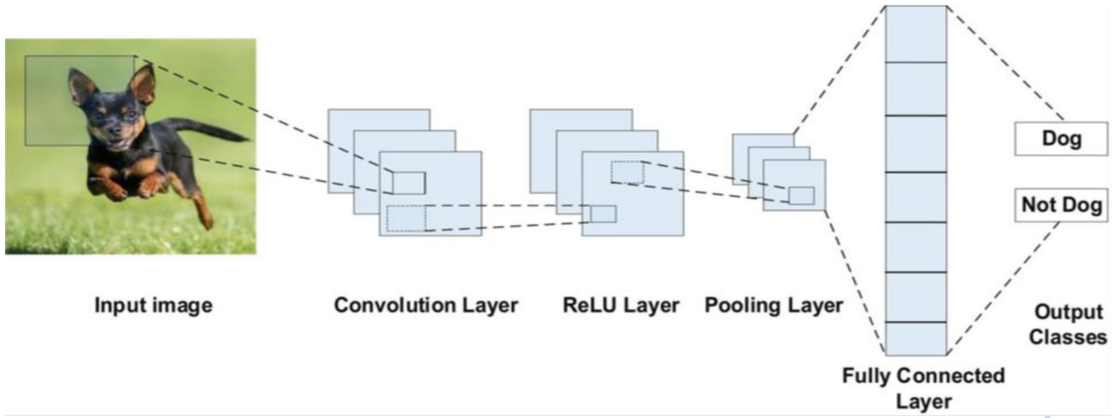


Figure 2.7: Convolutional Neural Networks Architecture for Image Classification[33]

Three elements are necessary for CNNs to function well: pooling, weight sharing, and local connectivity. Spatial local input patterns are extracted by hidden neurons in CNNs with the help of local connection. The deeper layers of a CNN can extract high-level semantic characteristics within a very broad receptive field, while the shallow levels of a CNN may extract low-level features, including textures and edges, within a relatively limited receptive field due to local connectedness. Weight sharing convolves all the visual fields with the same filtering mask, hence reducing the number of trainable parameters.

Weight sharing contributes to the achievement of translation invariance, which means that a significant feature may be extracted by the same filter regardless of its location, in addition to lowering the number of trainable parameters. CNNs frequently employ pooling procedures to minimize the dimensionality of input feature maps. They allow the latter layers of a CNN to focus on a wider variety of input patterns while simultaneously lowering the number of trainable parameters in the CNN. The majority of DL models employ max pooling, which is obtaining the greatest value in each subfield by applying a maximum filter to each feature map[37].

Benefits of Utilizing Convolutional Neural Networks (CNNs)

In the context of computer vision, employing CNNs has the following advantages over other conventional neural networks[33].

i. The weight sharing characteristic of CNN is the primary reason to take it into consideration. It lowers the number of trainable network parameters, which improves generalization and prevents overfitting.

ii. The model output becomes highly ordered and very dependent on the extracted features when the feature extraction layers and the classification layer are simultaneously learned.

iii. Compared to other neural networks, CNN implementation on a large scale is substantially simpler.

Layers of Convolutional Neural Networks (CNNs)

A number of layers also called multi-building blocks make up the CNN architecture. A detailed description of each layer of the CNN architecture, along with its purpose is discussed below[33].

i. Convolutional Layer: The convolutional layer is the most important part of the CNN model. It is comprised of variety of convolutional filters, also referred to as kernels. To create the output feature map, these filters are convolved with the input image. Below are the advantages of Convolutional Layer.

Sparse Connectivity: In fully connected neural networks, every neuron is connected to every other neuron in the layer below it. In CNNs, on the other hand, there are very few weights accessible between two neighboring layers. This makes this method memory-effective as there aren't many connections or weights needed, and there isn't much memory needed to store the weights. Additionally, matrix operations are computationally far more expensive compared to the CNN dot (.) operation.

Weight Sharing: As the entire weights function with every pixel in the input matrix, there are no assigned weights between any two neurons of contiguous layers in

CNN. Since it's not essential to learn different weights for every neuron, learning a single group of weights for the entire input will drastically reduce the amount of time and money needed for training.

ii. Pooling Layer: The subsampling of the feature maps is the pooling layer's primary function. The convolutional operations are followed to create these maps. Put otherwise, this method reduces the size of large-scale feature maps in order to produce smaller feature maps. Simultaneously, it preserves most of the dominating data or characteristics at each stage of the pooling process. The stride and the kernel are first size-assigned prior to the pooling procedure, just like in the convolutional operation. There are numerous kinds of pooling techniques that can be applied in different pooling layers. Tree pooling, gated pooling, average, min, max, global average pooling (GAP), and global max pooling are some of these techniques. The max, min, and GAP pooling are the most well-known and often used pooling techniques.

iii. Activation Function: The fundamental job of all kinds of activation functions in all kinds of neural networks is to map the input to the output. The computation of the weighted summation of the neuron input and its bias, if any, yields the input value. This indicates that by producing the appropriate output, the activation function decides whether or not to fire a neuron in response to a specific input. In CNN architecture, non-linear activation layers are used after all weighted layers, also known as learnable layers, such convolutional and fully connected layers. The mapping of input to output will be non-linear because to the activation layers' non-linear performance, which also enables the CNN to learn more complex tasks. The ability to distinguish is another crucial requirement for the activation function since it makes it possible to train the network using error back-propagation. The most popular activation function types in CNN and other deep neural networks are sigmoid, tanh, and ReLU.

iv. Fully Connected Layer: This layer is typically found at the very end of each CNN architecture. The professed fully connected technique is used in this layer, where every neuron is coupled to every other neuron in the layer above. It serves as the classifier for CNN. Being a feed-forward ANN, it operates on the same principles as a traditional multiple-layer perceptron neural network. The final pooling or convolutional layer provides the FC layer's input. Following flattening, the feature maps are converted into a vector, which is this input. The final CNN output is represented by the FC layer's output.

v. Loss Functions: Furthermore, the output layer, which is the last layer in the CNN architecture, achieves the final categorization. The predicted error produced by the CNN model over the training samples is computed at the output layer using a few loss functions. The discrepancy between the expected and actual output is shown by this error. It will then be optimized via CNN's learning process. Nevertheless, the error is computed using the loss function using two parameters. The first parameter is the CNN estimated output, also known as the prediction. The second parameter is the actual output, sometimes called the label. Different kinds of loss functions are used for different kinds of problems. These consist of the Hinge Loss Function, Euclidean Loss Function, and Cross-Entropy or Softmax Loss Function.

Optimization Techniques in Convolutional Neural Networks (CNNs)

In deep learning, the loss function idea indicates how poorly the model is doing at any given moment. It is necessary to train the network to perform better using this loss. In essence, the loss function must be minimized since greater performance of the model is indicated by a smaller loss function. Optimization is the process of minimizing or maximizing any mathematical expression[43]. Optimizers are algorithms or techniques that modify the weights and learning rate of a neural network in an effort to lower losses. The most accurate results can be obtained by minimizing the loss function by the use of optimization algorithms or strategies[44].

Optimizers in deep learning are algorithms that change the model's parameters in order to minimize a loss function. By iteratively adjusting weights and biases, they allow neural networks to learn from data. Stochastic Gradient Descent (SGD), Adam, and RMSprop are examples of common optimizers. To determine the ideal model parameters for increased performance, each optimizer includes unique update rules, learning rates, and momentum.

An optimization technique called an optimizer algorithm can help a deep learning model perform better. The accuracy and training speed of the deep learning model are significantly impacted by these optimization techniques, also known as optimizers. However, the first thing to consider is the definition of an optimizer. As you train the deep learning optimizer model, adjust the weights for each epoch and reduce the loss function. An optimizer is a function or algorithm that modifies the neural network's parameters, like learning rates and weights. As a result, it aids in raising accuracy and decreasing overall loss. With millions of parameters in most deep learning models, selecting the appropriate weights for the model is a difficult issue. It makes selecting an appropriate optimization more important.

Optimization can be done in a variety of ways, and each approach has benefits and drawbacks. Gradient descent is the most popular optimization technique used with neural networks. This technique entails continuously modifying the network's parameter values until performance gains. Different approaches and strategies can be used to optimize different kinds of problems. Gradient descent may not always be feasible, necessitating the employment of alternative techniques[46].

A neural network is trained when the parameters of its underlying model are adjusted so that the network can acquire the necessary skills to do the desired task. The loss function that will be used to assess the network's performance must be defined before the network can be trained. An algorithm known as a loss function is one that takes the network's output and uses a numerical value to indicate its performance. These parameters need to be specified with the intention of helping the network learn how to minimize the loss function value while doing the task effectively. The procedure of training a network involves modifying its parameter values in order to reduce the loss. A neural network's training process heavily relies on optimization. Numerous optimization algorithms have been suggested by recent research. There are benefits and drawbacks specific to each optimization algorithm[46].

Types of Optimizers that Minimize Loss Function

i. Gradient Descent: The most fundamental yet popular optimization approach is gradient descent. It plays a major role in algorithms for classification and linear regression. A gradient descent approach is also used in neural network backpropagation. An approach for first-order optimization called gradient descent depends on the loss function's first-order derivative. It determines how the weights should be changed in order to get the function to aminima. The loss is propagated from one layer to the next using backpropagation, and the model's parameters, or weights, are adjusted based on the losses in order to minimize the loss[44].

   Algorithm of gradient descent: $\theta = \theta - \alpha \cdot \nabla J(\theta)$

   Advantages of gradient descent:

   a. Stress-free computation.

b. Easy implementation.

c. Easy to understand.

Disadvantages of gradient descent:

a. It may trap at local minima.

b. After calculating the gradient throughout the entire dataset, weights are adjusted. Therefore, it could take years for this to converge to the minima if the dataset is too big.

c. It needs large memory to compute gradient on the entire dataset.

ii. Stochastic Gradient Descent (SGD): It is a Gradient Descent variation. It makes an effort to adjust the model's parameters more regularly. This involves changing the model parameters following the loss calculation for every training set. Therefore, instead of updating the model parameters once like in Gradient Descent, SGD will update them 1000 times in a single dataset cycle if the dataset has 1000 rows[44].

Algorithm of stochastic gradient descent: $\theta=\theta-\alpha\cdot\nabla J(\theta;x(i);y(i))$, where $\{x(i), y(i)\}$ are the training examples. The model's parameters have a high variation and fluctuate in the loss functions at different intensities because they are updated often.

Advantages stochastic gradient descent:

a. Regular update of the model's parameters, hence, comes together in less time.

b. Necessitates less memory as there is no need to store values of loss functions.

c. It may get new minima.

Disadvantages stochastic gradient descent:

a. Increased difference in model parameters.

b. It may shoot after reaching global minima.

c. In order to get the same convergence as gradient descent, it must slowly decrease the value of learning rate.

iii. Mini Batch Stochastic Gradient Descent (MB-SGD): The SGD algorithm's large time complexity is solved by the MB-SGD algorithm, which is an extension of the SGD algorithm. To compute the derivative, the MB-SGD technique uses a batch of points or a portion of the dataset. After a certain number of iterations, it is found that the derivative of the loss function for MB-SGD is nearly identical to the derivative of the loss function for Gradient Descent. However, MB-SGD requires more iteration to reach minima than Gradient Descent, and it also comes with a higher computation cost. The derivate of loss for a batch of points determines how the weight is updated. Because the derivative in the MB-SGD example is not necessarily towards minima, the updates are substantially noisier. The dataset is divided into many batches by MB-SGD, and the parameters are changed after each batch[43].

Algorithm of MB-SGD: $\theta=\theta-\alpha\cdot\nabla J(\theta;\ B(i))$, where $\{B(i)\}$ are the batches of training examples.

Advantages Mini Batch Stochastic Gradient Descent

a. By lowering the variance in the parameter updates, it can eventually result in a much more stable and improved convergence.
b. It can take advantage of highly improved matrix optimizations seen in modern deep learning packages, which greatly increase the efficiency of computing the gradient with respect to a mini-batch.
c. Mini-batch sizes usually range from 50 to 256, though this can change depending on the problem being handled and the application.
d. These days, while training a neural network, the preferred approach is

usually mini-batch gradient descent[47].

Challenges Faced While Using Gradient Descent and Its Variants

a. Selecting the right learning rate might be challenging. A learning rate that is too low results in incredibly slow convergence, or tiny baby steps toward identifying the ideal parameter values that minimize loss and locate that valley, which directly impacts the training time, which is excessively long. On the other hand, an excessively high learning rate might impede convergence and result in the loss function varying near the minimum or even diverging.

b. Furthermore, all parameter updates are subject to the same learning rate. We might not want to update every feature to the same degree if the data is poor and the features have widely disparate frequencies. Instead, we might wish to update the seldom occurring features more extensively.

c. Staying out of the multiple sub-optimal local minima that arise from the minimization of extremely non-convex error functions, which are typical of neural networks, is another big challenge. Sack spots, or the places where one dimension slopes up and another down, are where difficulty actually originates, not local minima. Due to the gradient's near-zero values in both dimensions, SGD is infamously difficult to escape from these saddle spots, which are typically encircled by a plateau of constant error[47].

Various Algorithms which are used to Further Optimize Gradient Descent

i. Momentum: Momentum was developed in order to ease the convergence and lower the large variance in SGD. It lessens the fluctuation in the irrelevant direction and quickens the convergence towards the relevant direction. In this approach, there is an additional hyperparameter called momentum, which is represented by 'γ'[44].

Algorithm of momentum: $V(t)=\gamma V(t-1)+\alpha.\nabla J(\theta)$. The weights are updated by $\theta=\theta-V(t)$. The momentum term γ is usually set to 0.9 or a similar value.

Advantages of Momentum

a. It lowers oscillations and high variance of the parameters.

b. It converges more quickly than gradient descent.

Disadvantages of momentum

a. One additional hyper-parameter that needs to be carefully chosen manually and precisely is included.

ii. Nesterov Accelerated Gradient: Although momentum can be a useful tool, an algorithm may miss the local minima and keep increasing if the momentum is too high. Therefore, the Nesterov Accelerated Gradient algorithm was created to address this problem. It is an approach that looks ahead. By adjusting the weights using $\gamma V(t-1)$ to roughly determine the future location, $\theta-\gamma V(t-1)$ can be obtained[44]. $V(t)=\gamma V(t-1)+\alpha.\ \nabla J(\ \theta-\gamma V(t-1)\ )$ and then update the parameters using $\theta=\theta-V(t)$.

Advantages of Nesterov Accelerated Gradient

a. It doesn't miss the local minima.

b. It slows down if minimas are occurring.

Disadvantage of Nesterov Accelerated Gradient

a. Hyperparameter is still required to be chosen manually.

iii. Adaptive Gradient (AdaGrad): One major drawback of all the optimizers explained is that the learning rate is constant for all parameters and for each cycle. This optimizer modifies the learning rate. It modifies the learning rate 'η' for each parameter and at every time step 't'. It is a type second order optimization algorithm. It operates on the derivative of an error function[44].

Advantages of Adaptive Gradient

a. For each training parameter, the learning rate changes.

b. There is no need to tune the learning rate manually.

c. It has the ability to train on sparse data.

Disadvantages of Adaptive Gradient

a. Calculating the second order derivative is computationally expensive.

b. The continuous decrease in learning rate leads to slow training.

iv. AdaDelta: It is an AdaGrad extension that aims to fix the issue of its declining learning rate. Adadelta restricts the window of accumulated past gradients to a set size w, as opposed to aggregating all previously squared gradients. Instead of using the total of all gradients, an exponentially moving average is utilized[44]. The advantages of Adadelta is that learning rate doesn't decay and the training doesn't stop and the disadvantages it is computationally expensive.

v. RMSprop: RMSprop is similar to the first update vector of Adadelta. Additionally, RMSprop divides the learning rate by the average of squared gradients, which decays exponentially. Hinton recommends setting γ to 0.9, and 0.001 is a suitable default value for the learning rate η. At the same time, Adagrad's drastically declining learning rates prompted the independent development of RMSprop and Adadelta[43].

vi. Adam (Adaptive Moment Estimation): Adam deals with first- and second-order momentums. The Adam was designed with the intuitive idea that we should roll more slowly to allow for a more thorough search rather than rolling as quickly as possible to clear the minimum. Adam maintains an exponentially decaying average of past gradients M(t), in addition to an exponentially decaying average of past squared gradients, like AdaDelta. The values of the first moment M(t) represent the gradients' mean, and the second moment V(t) represents their uncentered variance[44].

Advantages of Adam

a. This approach is too fast and converges quickly.
b. It corrects vanishing learning rate and high variance problem.

Disadvantages of Adam

a. It is computationally expensive.

CNN Architectures for Image Classification

Deep learning frameworks that are most widely used are CNN architectures. While CNNs have demonstrated impressive results in solving image identification problems and offering unprecedented scalability and precision, all CNNs are not made equal[48]. While it is feasible to create a CNN from the ground up, architectures that have been created and made available to the public can also be utilized. Additionally, some of these networks include pre-trained models that are simple to modify for a given use case[49].

Types of CNN Architectures

The following is a list of various categories of CNN architectures

i. LeNet: The initial CNN architecture was called LeNet. Yann LeCun, Corinna Cortes, and Christopher Burges created it in 1998 to solve handwritten digit recognition challenges. One of the earliest CNNs to be effective was LeNet, which is frequently referred to as the "Hello World" of deep learning. It is among the most popular and ancient CNN architectures and it has proven effective for a variety of applications, including handwritten digit recognition. Multiple convolutional and pooling layers make up the LeNet architecture, which is followed by a fully-connected layer. The model consists of two fully linked layers after five convolutional layers. CNNs first appeared in deep learning for computer vision applications with LeNet. However, the vanishing gradients issue prevented LeNet from training effectively. In order to address this problem, convolutional layers employ a shortcut connection layer called max-pooling to minimize the spatial dimension of images. This helps to prevent overfitting and improves CNN training efficiency. The architecture of LeNet-5 is shown in the diagram below.

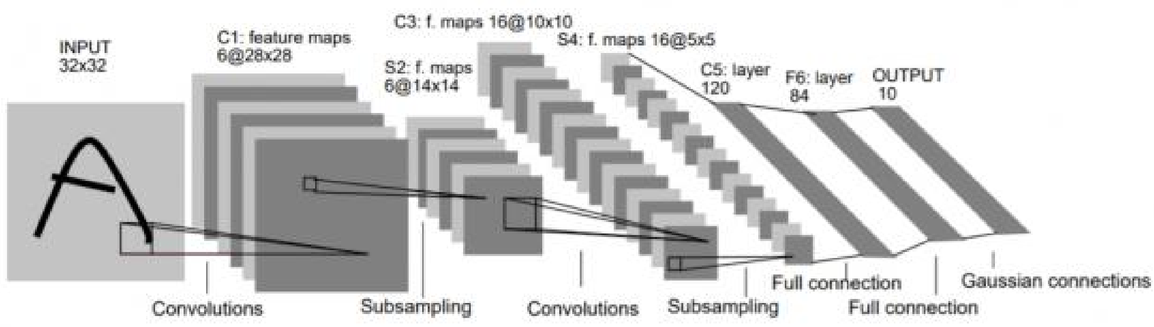


Figure 2.8: LeNet Convolutional Neural Networks Architecture[48]

As a straightforward yet effective model, the LeNet CNN has been applied to a number of tasks, including face detection, traffic sign recognition, and handwritten digit recognition. LeNet was created more than 20 years ago, yet its architecture is still applicable and in use today.

ii. AlexNet: The deep learning architecture that made CNN prominent is called AlexNet. Geoff Hinton, Ilya Sutskever, and Alex Krizhevsky developed it. Though it was larger, deeper, and included convolutional layers stacked on top of one another, the AlexNet network's architecture was quite similar to that of LeNet. In order to win the 2012 ImageNet Large Scale Visual Recognition Challenge (ILSVRC), AlexNet, the first large-scale CNN, was employed. When the AlexNet architecture was first published, it produced state-of-the-art results and was intended to be utilized with large-scale picture datasets. AlexNet consists of three fully connected layers, two dropout layers, and five convolutional layers that combine max-pooling and other layers. Relu serves as the activation function in every layer. The output layer uses Softmax as its activation function. This architecture has over 60 million parameters in total[48].

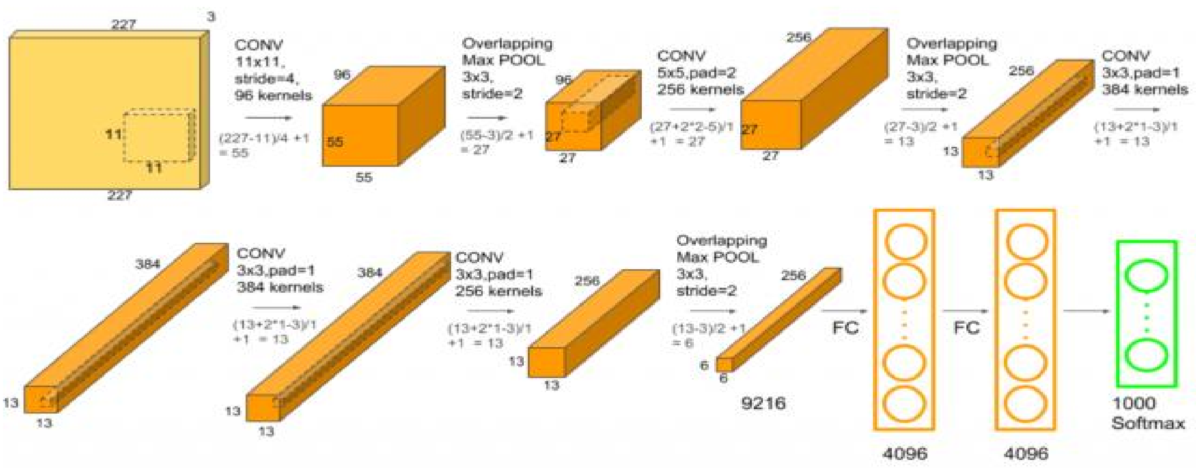


Figure 2.9: AlexNet Convolutional Neural Networks Architecture[48]

iii. ZF Net: The CNN architecture known as ZFnet combines CNNs with fully-connected layers. Rob Fergus and Matthew Zeiler created ZF Net. It was the 2013 winner of the ILSVRC. With only 1000 images per class, the network achieves top accuracy on the ILSVRC 2012 classification test, outperforming AlexNet despite having comparatively less parameters. By adjusting the architecture hyperparameters, namely the size of the middle convolutional layers and the stride and filter size on the first layer, it performed better than AlexNet. The Zeiler and Fergus model, which was trained using the ImageNet dataset, serves as its foundation. The convolutional layer, max-pooling layer (downscaling), concatenation layer, convolutional layer with linear activation function, and stride one are the seven layers that make up the ZF Net CNN architecture. Dropout is performed before the fully connected output for regularization reasons. ZFnet is computationally more effective than AlexNet by introducing an approximation inference step through deconvolutional layers in the middle of CNNs[48].

iv. GoogLeNet: GoogLeNet is the CNN architecture utilized by Google to win ILSVRC 2014 classification task. Jeff Dean, Christian Szegedy, Alexandro Szegedy, and others developed it. It has been demonstrated to have a significantly lower error rate when compared to the winners of the previous ILSVRC competitions, AlexNet (2012 winner) and ZF-Net (2013 winner). Compared to VGG (the 2014 runner-up), the error is substantially smaller in terms of error rate. It uses several different approaches, such as global average pooling and 1×1 convolution, to create deeper architecture. The CNN design of GoogleNet is computationally costly. It features shortcut connections between the first two convolutional layers before adding additional filters in later CNN layers, and it employs heavy unpooling layers on top of CNNs to remove spatial redundancy during training, hence reducing the number of parameters that need to be learned. The Street View House Number (SVHN) digit recognition challenge is a real-world application/example of GoogLeNet CNN architecture and is frequently used as a stand-in for roadside object detection[48]. The simplified block diagram of GoogleNet CNN architecture is shown in Figure 2.10.

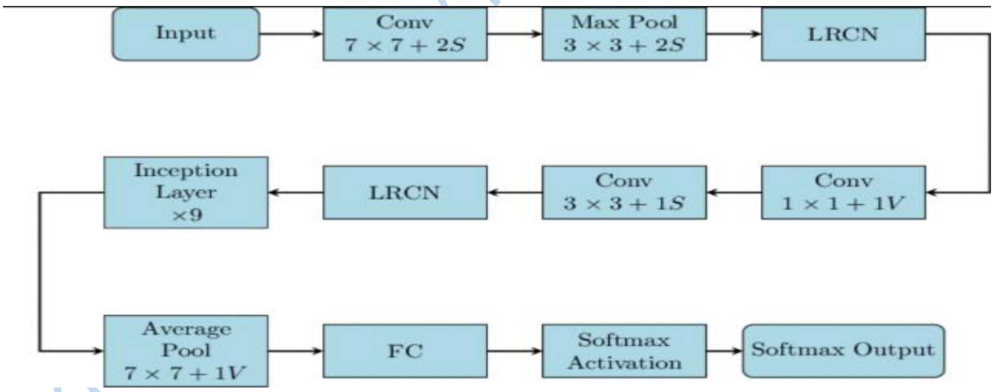


Figure 2.10: GoogLeNet Convolutional Neural Networks Architecture[48]

v. VGGNet: The CNN architecture known as VGGNet was created at Oxford University by Andrew Zisserman, Karen Simonyan, and others. With up to 95 million parameters, VGGNet is a 16-layer CNN that was trained on more than one billion images (1000 classes). With 4096 convolutional features, it can process huge input images up to 224 by 224 pixels in size. For the majority of image classification tasks when input images have a size between 100 x 100 pixels and

350 x 350 pixels, CNN architectures like GoogLeNet (AlexNet architecture) perform better than VGGNet since CNNs with such large filters are costly to train and require a lot of data. The ILSVRC 2014 classification task is one of the real-world applications/examples of VGGNet CNN architecture, which was also won by GoogleNet CNN architecture. Because it can be used for a wide range of tasks, including object detection, the VGG CNN model is highly applicable and provides a solid foundation for many computer vision applications. It is also computationally efficient. Its deep feature representations are used across multiple neural network architectures like YOLO, SSD, etc[48]. The typical VGG16 network architecture diagram is shown in the diagram below.

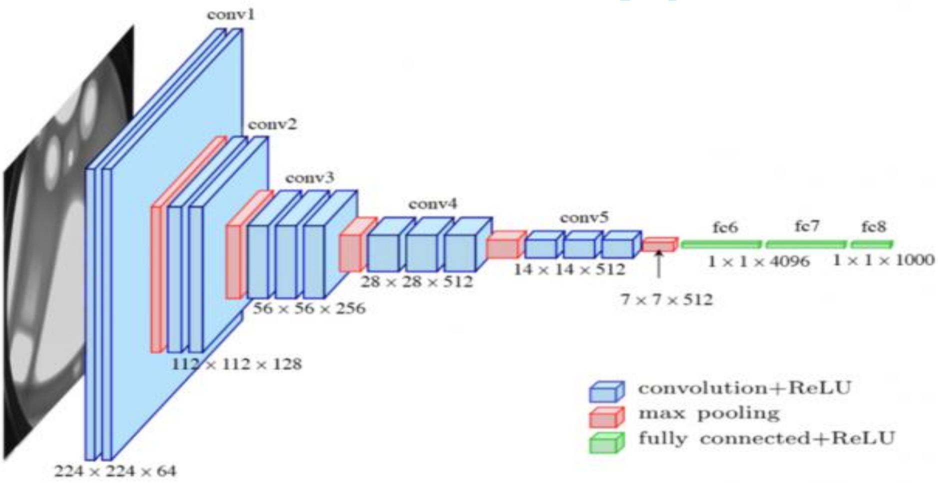


Figure 2.11: VGGNet Convolutional Neural Networks Architecture[48]

vi. ResNet: Kaiming He et al. created the CNN architecture known as ResNet, which allowed them to win the ILSVRC 2015 classification assignment with a top-five error rate of just 15.43%. With over a million parameters and 152 layers, the network is considered deep even for CNNs. It was trained using the ILSVRC 2015 dataset over the course of more than 40 days on 32 GPUs. The Microsoft Research Asia team used ResNet in 2016 and 2017, respectively, to demonstrate that CNNs can be successfully applied to natural language processing problems such as sentence completion and machine comprehension. Traditionally, CNNs are used for image classification tasks with 1000 classes. Microsoft's machine comprehension system is one of the real-world uses of the ResNet CNN architecture; it generates responses for over 100,000 questions across 20 categories using CNNs. ResNet, the CNN architecture, has a computationally efficient design that allows it to be scaled up or down to meet GPU processing capability.

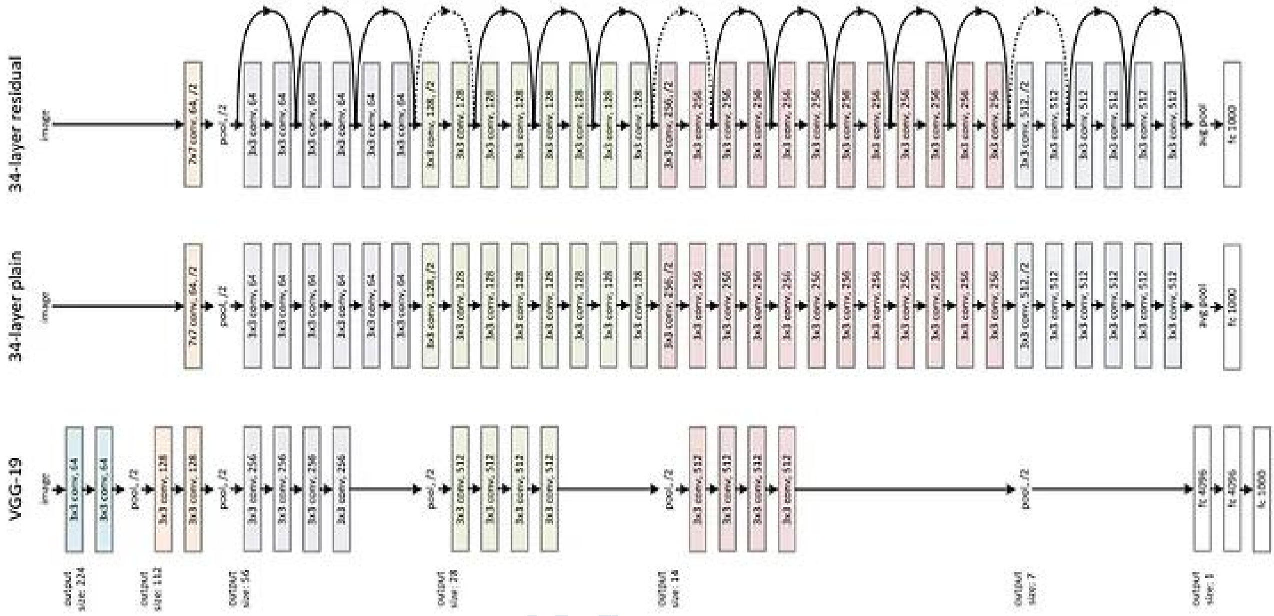


Figure 2.12: Residual Network Architecture[50]

vii. MobileNets: MobileNets are CNNs that can be installed on a mobile device and used to quickly identify objects or classify images. The creators of MobileNets are Andrew G. Trillion and colleagues. Because they are often fairly compact CNN structures, embedded devices like smartphones and drones can easily operate them in real-time. Due to the architecture's flexibility, CNNs with 100–300 layers have been used to test it, and results show that it performs better than competing architectures like VGGNet. CNNs integrated inside Android phones to run Google's Mobile Vision API, which can automatically recognize labels of well-known items in images, are examples of real-world MobileNets CNN architecture[48].

viii. GoogLeNet_DeepDream: GoogLeNet_DeepDream CNN architecture was created by Christopher Olah, Alexander Mordvintsev, and others. It creates graphics based on CNN features by utilizing the Inception network. The architecture is frequently utilized with the ImageNet dataset to produce psychedelic visuals or abstract artworks using human imagination[48].

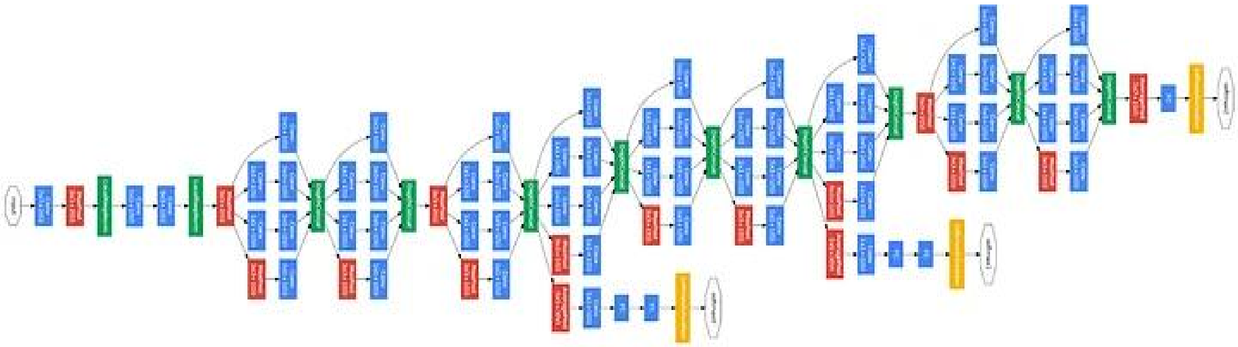


Figure 2.13: GoogLeNet Architecture[50]

ix. Squeeze Net: With its eighteen deep layers, it can classify images of objects into 1000 different categories, such as a pencil, a keyboard, a mouse, and many more creatures. With the same accuracy, SqueezeNet can be three times faster and 500 times smaller than AlexNet[51].

x. DenseNet: During the CVPR Conference, Densely Connected Convolutional Networks, developed by Gao Huang, Zhuang Liu, and colleagues in 2017, was referred to as DenseNet[7]. It has received over 2000 citations and been awarded the best article honor. Each layer of a conventional convolutional network has n connections. However, DensetNet has n(n+1)/2 connections total because of its feed-forward architecture[51].

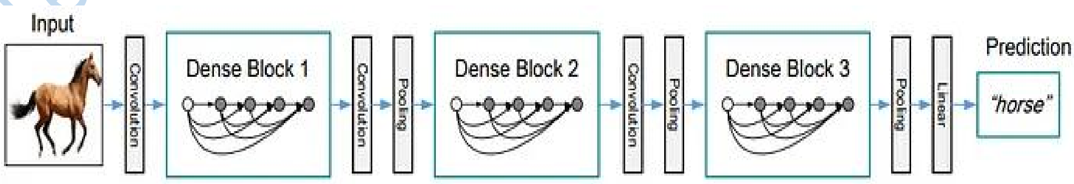


Figure 2.14: DenseNet Architecture[50]

xi. Shuffile Net: Designed for mobile devices, the 173 deep layers CNN architecture had a CPU power of 10-150 MFLOPs, making it extremely effective. It may obtain a lower top-1 error (absolute 7.8%) on Image Net classification than Mobile Net system[51].

xii. ENet (Efficient Neural Network): Pixel-wise semantic segmentation in real-time is made possible by the ENet. With up to eighteen times quicker processing speed, seventy-nine times fewer FLOPs needed, and 79 times less parameters, ENet provides comparable or better accuracy than earlier models. ENet is the quickest model in semantic segmentation[51].

Table 2.1: Brief Overview of CNN Architectures

Architecture	Finding	Depth	Dataset	Error Rate	Size	Year
AlexNet	Make use of Dropout and Rectified Linear Unit	8	ImageNet	16.4	227 × 227 × 3	2012
NIN	It has new layer, called ‘mlpconv’, which uses Global Average Pooling	3	CIFAR-10, CIFAR-100, MNIST	10.41, 35.68, 0.45	32 × 32 × 3	2013
ZfNet	Visualization idea of middle layers	8	ImageNet	11.7	224 × 224 × 3	2014
VGG	It has increased depth and small filter size	16, 19	ImageNet	7.3	224 × 224 × 3	2014
GoogLeNet	It has increased depth, block concept, dissimilar kernel size, and concept of concatenation	22	ImageNet	6.7	224 × 224 × 3	2015
Inception-V3	It uses small kernel size and has better feature representation	48	ImageNet	3.5	229 × 229 × 3	2015
Highway	It presented the concept of multipath	19, 32	CIFAR-10	7.76	32 × 32 × 3	2015
Inception-V4	It has divided transform and integration concepts	70	ImageNet	3.08	229 × 229 × 3	2016
ResNet	It is very robust against overfitting due to symmetry mapping-based skip links	152	ImageNet	3.57	224 × 224 × 3	2016
Inception-ResNetv2	It introduced the concept of residual links.	164	ImageNet	3.52	229 × 229 × 3	2016
FractalNet	the concept of Drop-Path is introduced as regularization	40,80	CIFAR-10 CIFAR-100	4.60 18.85	32 × 32 × 3	2016
WideResNet	It decreases the depth and increases the width	28	CIFAR-10 CIFAR-100	3.89 18.85	32 × 32 × 3	2016
Xception	A depthwise convolution is followed by a pointwise convolution	71	ImageNet	0.055		2017
Residual attention neural	It presented the attention technique	452	CIFAR-10, CIFAR-100	3.90, 20.4	40 × 40 × 3	2017

network						
Squeeze-and-excitation networks	It modeled interdependencies between channels	152	ImageNet	2.25	229 × 229 × 3 224 × 224 × 3 320 × 320 × 3	2017
DenseNet	It has blocks of layers. i.e. layers connected to each other	201	CIFAR-10, CIFAR-100 ImageNet	3.46, 17.18, 5.54	224 × 224 × 3	2017
Competitive squeeze and excitation network	Residual and identity mappings are both used to rescale the channel	152	CIFAR-10, CIFAR-100	3.58 18.47	32 × 32 × 3	2018
MobileNet-v2	It has inverted residual structure	53	ImageNet	–	224 × 224 × 3	2018
CapsuleNet	It pays attention to special associations between feature	3	MNIST	0.00855	28 × 28 × 1	2018
HRNetV2	It has high-resolution representations	–	ImageNet	5.4	224 × 224 × 3	2020

CNN Architectures[33]

2.1.20 Methods of Creating and Training Deep Learning Models

Robust deep learning models can be built using a variety of techniques. These methods include of dropout, training from scratch, transfer learning, and learning rate decay[52].

i. Learning Rate Decay: The amount of change the model undergoes in response to the estimated error each time the model weights are adjusted is determined by the learning rate, a hyperparameter that defines the system or establishes operating conditions for it before the learning process. Overly high learning rates might lead to unstable training procedures or the acquisition of a less-than-ideal weight set. Too low of a learning rate can result in a training process that is too long and can become stalled. The simplest and most popular ways to modify the learning rate during training are those that gradually lower the learning rate.

ii. Transfer Learning: This approach entails fine-tuning a model that has already been trained; it needs an interface to a preexisting network's internal workings. Users first feed new data with previously unidentified classifications into the network that already exists. Once the network has been adjusted, new jobs can be completed with more precise classification skills. Compared to other methods, transfer learning has the advantage of requiring far less data, which cuts down the computation time to minutes or hours.

iii. Training from Scratch: In order to use this strategy, a developer must assemble a sizable labeled dataset and put up network architecture with feature and model learning capabilities. This method is particularly helpful for newly developed applications and those with a big number of output categories. All things considered, it is a less popular method because training takes days or weeks and necessitates excessive volumes of data.

iv. Dropout: This technique randomly removes units and their connections from the neural network during training in an effort to address the issue of overfitting in networks with a large number of parameters. The dropout technique has been shown to enhance neural networks' performance on supervised learning tasks, including speech recognition, document categorization, and computational biology.

2.1.21 Applications of Deep Learning

Natural language processing (NLP), reinforcement learning, and computer vision might be considered the three main areas of application for deep learning[36].

i. Computer Vision: Deep learning models in computer vision can help machines recognize and comprehend visual input. The following are a few of the primary computer vision applications of deep learning:

ii. Object Detection and Recognition: Automated tasks like robotics, self-driving cars, and surveillance are made possible by deep learning models that can detect and recognize objects in images and videos.

iii. Image Classification: Images can be categorized using deep learning models into groups like people, pets, and buildings. Application area includes image retrieval, quality control, and medical imaging.

iv. Image Segmentation: By segmenting images into distinct regions, deep learning models can be used to recognize particular features inside images.

v. Natural Language Processing (NLP): The Deep Learning approach in NLP can help machines comprehend and produce human language. Among the principal uses of deep learning in NLP are:

vi. Automatic Text Generation: With the help of these trained models, essays and other new writing can be automatically generated by deep learning models that have learned a corpus of text.

vii. Language Translation: Communicating with people who speak various languages is made possible by deep learning models that translate text between languages.

viii. Sentiment analysis: It is possible to ascertain whether a text is good, negative or neutral by using deep learning models to analyze the sentiment of the text. Applications like social media monitoring, customer service and political analysis leverage this.

ix. Speech recognition: With the ability to identify and transcribe spoken words, deep learning models can be used for voice-activated devices, speech-to-text conversion, and voice search.

x. Reinforcement Learning: Deep learning is used in reinforcement learning to train agents to behave in a way that maximizes a reward. The following are some of the primary uses of deep learning in reinforcement learning:

xi. Game playing: In games like Atari, Chess, and Go, deep reinforcement learning models have demonstrated superior performance compared to human specialists.

xii. Robotics: It is possible to educate robots to carry out intricate tasks including manipulating, navigating, and grabbing items by using deep reinforcement learning models.

xiii. Control systems: Supply chain optimization, traffic management, and power grids are a few examples of complex systems that can be controlled by deep reinforcement learning models.

2.1.22 Challenges in Deep Learning

Although deep learning has made great strides in many areas, there are still certain issues that need to be resolved[36]. The following are a few of the primary obstacles in deep learning:

i. Data Availability: To learn from, a lot of data is needed. Acquiring as much training data as possible is crucial for deep learning applications.

ii. Computational Resources: It is computationally expensive to train the deep learning model because it needs specialized hardware, such as GPUs and TPUs.

iii. Time Consuming: Working with sequential data might take a very long time, even in days or months, depending on the computational resources available.

iv. Interpretability: Deep learning models are intricate and operate in a mysterious manner. The outcome is really hard to understand.

v. Overfitting: Repetition of training causes the model to become excessively specialized for the training set, which results in overfitting and reduced performance on fresh data.

2.1.23 Advantages of Deep Learning

i. High Accuracy: Deep Learning algorithms are capable of achieving cutting edge results in a variety of applications, including natural language processing and picture recognition.

ii. Automated Feature Engineering: Without the need for human feature engineering, deep learning algorithms can automatically find and learn useful features from data.

iii. Scalability: Massive volumes of data can be used to train deep learning models, which can handle complex and large-scale datasets.

iv. Flexibility: Deep Learning models are versatile tools that may be used for a wide range of activities and data kinds, including text, speech, and images.

v. Continual Improvement: As more data becomes available, deep learning models' performance can continuously improve[36].

2.1.24 Disadvantages of Deep Learning

i. High Computational Requirements: To train and optimize, deep learning models need a lot of data and processing power.

ii. It requires large amounts of labeled data: For training, deep learning models frequently need a lot of labeled data, which can be costly and time-consuming to get.

iii. Interpretability: It can be difficult to interpret deep learning models, which makes it tough to comprehend how they arrive at judgments.

iv. Overfitting: Sometimes, overfitting to the training set causes deep learning models to perform poorly on fresh, untrained data.

v. Black-Box Nature: Deep Learning models are frequently regarded as "black boxes," which makes it challenging to comprehend how they operate and make predictions[36].

2.1.25 Evaluation Metrics Used in Deep Learning

The selection of evaluation metrics in deep learning tasks is essential to obtaining an optimal classifier. They are used in two primary phases of a typical data classification process: testing and training. During the training phase, it is used to optimize the classification algorithm. This means that the evaluation metric is used, for example, as a discriminator that can produce an exceptionally accurate forecast of future evaluations linked to a particular classifier, in order to discriminate and choose the optimal answer. For now, the evaluation metric is used to gauge the developed classifier's effectiveness, for example, as an assessor during the hidden data model testing phase. The following equations define TN and TP, respectively, as the number of accurately classified negative and positive examples. Furthermore, the definitions of FN and FP are the quantity of incorrectly identified positive and negative cases, respectively[33]. The following is a list of some of the most well-known evaluation measures.

i. Accuracy: Accuracy computes the ratio of correct predicted classes to the total number of samples evaluated.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \qquad \text{Equation (2.1)}$$

ii. Sensitivity or Recall: It is used to compute the fraction of positive patterns that are correctly classified.

$$\text{Sensitivity} = \frac{TP}{TP + FN} \qquad \text{Equation (2.2)}$$

iii. Specificity: It is used to compute the fraction of negative patterns that are correctly classified.

$$\text{Specificity} = \frac{TN}{FP + TN} \qquad \text{Equation (2.3)}$$

iv. Precision: It is used to compute the positive patterns that are correctly predicted by all predicted patterns in a positive class.

$$\text{Precision} = \frac{TP}{TP + FP} \qquad \text{Equation (2.4)}$$

v. F1-Score: It is used to estimate the harmonic average between recall and precision rates.

$$\text{F1score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \qquad \text{Equation (2.5)}$$

vi. J Score: This metric is also called Youdens J statistic.

$$\text{Jscore} = \text{Sensitivity} + \text{Specificity} - 1 \qquad \text{Equation (2.6)}$$

vii. False Positive Rate (FPR): This evaluation metric refers to the likelihood of a false alarm ratio.

$$\text{FPR} = 1 - \text{Specificity} \qquad \text{Equation (2.7)}$$

viii. Area Under the ROC Curve (AUC): One popular ranking type metric is AUC. It is used to create the best possible learning model and to compare different learning algorithms. Unlike probability and threshold measurements, the full classifier ranking performance is shown by the AUC value. The following formula is used to calculate the AUC value for two-class problem.

$$\text{AUC} = \frac{S_p - n_p(n_n + 1)/2}{n_p n_n} \qquad \text{Equation (2.8)}$$

Here, Sp represents the sum of all positive ranked samples. The number of negative and positive samples is denoted as n_n and n_p, respectively. The AUC value was practically and theoretically proven in comparison to the accuracy metrics, which makes it a valuable tool for classifier performance evaluation via classification training and for determining an optimal solution. The AUC performance was excellent when taking the evaluation and discrimination methods into account. However, when discriminating a large number of produced solutions, the AUC computation is largely cost-effective for multiclass situations. Furthermore, the Hand and Till AUC model yields an O(|C|2 n log n) time complexity for calculating the AUC, while Provost and Domingo's AUC model yields an O(|C| n log n) time complexity.

2.1.26 Deep Learning Frameworks and Datasets

In the past few years, numerous DL frameworks and datasets have been created. A number of frameworks and libraries have also been utilized to speed up the process and produce quality output. The training procedure has been made simpler by their utilization. The most popular libraries and frameworks are listed in Table 2.2. Github users' star ratings indicate that TensorFlow is the most efficient and user-friendly tool. It can function across multiple platforms. Github is one of the largest websites for software hosting; a project's standing on the site is indicated by its "stars". Additionally, a number of additional benchmark datasets are used for various DL tasks[33]. Table 2.3 has a list of some of these.

Table 2.2: List of the Most Common Frameworks and Libraries

Framework	License	Preogramming Language	Year	Website
TensorFlow	Apache 2.0	C++ & Python	2015	https:// www. tenso rflow. org/
Keras	MIT	Python	2015	https:// keras. io/
Caffe	BSD	C++	2015	http:// caffe. berke leyvi sion. org/
MatConvNet	Oxford	MATLAB	2014	http:// www. vlfeat. org/ matco nvnet/
MXNet	Apache 2.0	C++	2015	https:// github. com/ dmlc/ mxnet
CNTK	MIT	C++	2016	https:// github. com/ Micro soft/ CNTK
Theano	BSD	Python	2008	http:// deepl earni ng. net/ softw are/ theano/
Torch	BSD	C & Lua	2002	http:// torch. ch/
DL4j	Apache 2.0	Java	2014	https:// deepl earni ng4j. org/
Gluon	AWS Microsoft	C++	2017	https:// github. com/ gluon- api/ gluon- api/
OpenDeep	MIT	Python	2017	http:// www. opend eep. org/

Frameworks and Libraries[33]

Table 2.3: Benchmark Datasets

Dataset	Number of Classes	Applications	Website
ImageNet	1000	Image classification, object localization, object detection, etc.	http://www.image- net.org/
CIFAR10/100	10/100	Image classification	https://www.cs.toronto.edu/~kriz/cifar.html
MNIST	10	Classification of handwritten digits	http://yann.lecun.com/exdb/mnist/
Pascal VOC	20	Image classification, segmentation, object detection	http://host.robots.ox.ac.uk/pascal/VOC/voc2012/
Microsoft COCO	80	Object detection, semantic segmentation	https://cocodataset.org/#home
YFCC100M	8M	Video and image understanding	http://projects.dfki.unikl.de/yfcc100m/
YouTube-8M	4716	Video classification	https://research.google.com/youtube8m/
UCF-101	101	Human action detection	https://www.crcv.ucf.edu/data/UCF101.php
Kinetics	400	Human action detection	https://deepmind.com/research/open-source/kinetics
Google Open Images	350	Image classification, segmentation, object detection	https://storage.googleapis.com/openimages/web/index.html
CalTech101	101	Classification	http://www.vision.caltech.edu/Image_Datasets/Caltech101/
Labeled Faces in the Wild	–	Face recognition	http://vis- www.cs.umass.edu/lfw/
MIT-67 scene dataset	67	Indoor scene recognition	http://web.mit.edu/torralba/www/indoor.htm

Benchmark Datasets[33]

2.2 Related Works

Brain tumors can be detected and graded using a variety of imaging modalities, including MRI with the help of widespread application of machine learning (ML) and deep learning (DL) techniques[13]. An overview of various current research projects along with the findings is provided in this section. This section is divided into two: related article on brain tumor classification and related article on input size limitation.

2.2.1 Brain Tumor Classification

This study examined the use of modern deep learning techniques for the automatic classification of magnetic resonance images. Here, an evaluation of the state of the art and upcoming advancements required to bring MRI-based brain tumor categorization techniques into regular clinical practice were also given[53].

This research presents a unique paradigm for the identification and categorization of brain tumors. The core idea is to take a small class-unbalanced collected dataset and use it to create a large synthetic MRI image dataset that represents the typical pattern of the brain MRI images. A deep model is then trained for detection and classification using the output dataset. Precisely, two kinds of deep models were used. Generative model was employed to capture the distribution of the significant features in a group of small class-unbalanced brain MRI images. The generative model can then create any number of brain MRI images for each class using this distribution. As a result, a few unbalanced datasets can be automatically converted by the system into a larger balanced dataset. In order to identify brain cancers in MRI images, the second model is a classifier that was developed using a sizable, well-balanced dataset. With an overall detection accuracy of 96.88%, the proposed framework shows great promise as a low-cost, accurate method for detecting brain tumors[54].

This study suggested a method for classifying glioma brain tumors non-invasively using an altered version of AlexNet CNN. Whole-brain MRI images were used for the classification procedure, and labels were applied at the image level rather than the pixel level. The experimental findings demonstrated that the method's accuracy of 91.16% was a respectable performance[55].

This work covered the efficient use of deep learning CNN for the classification of MR images into normal and pathological cells. Matlab is used in this method for preprocessing, classification, and feature extraction[56].

The application of convolutional neural networks was examined in this study. Three classifiers were used to classify the images: random forest, fully connected neural network (FCNN), and CNN. Two convolutional layers, each with 64 filters measuring 5 x 5, MaxPool layers, two fully connected layers with 800 neurons, and a SoftMax layer serving as the output layer were the configurations of the CNN[57].

In order to identify the CT brain image, this work developed 2D and 3D-dimensional convolutional deep learning models with seven layers. In each layer, convolution and subsampling were used. In the best modeling, the average classification precision was 87.6%. Nevertheless, these studies did not take into account the CNN's hyper-parameters, which are crucial to guaranteeing the learning algorithm's optimal performance[58].

The study uses hybrid feature selection in conjunction with ensemble classification to diagnose brain tumors. To derive the decision rules, a wrapper technique based on GANNIGMAC, decision trees, and bagging C is employed. Hybrid feature selection, which combines GANNIGMAC + MRMR C+ Bagging C + Decision Tree was used to simplify the decision rules[59].

In this study, glioma brain tumor prediction was achieved by the application of machine learning techniques. ROI was found for every slice in order to get the region histogram.

To extract the features and train the model, a pretrained CNN technique was employed. AlexNet, which was both pretrained and trained, had the best classification accuracy, which did not go over 46%. For two classifications, the histogram features achieved 68.5% accuracy[60].

In this study, a technique for segmenting and categorizing MRI brain images into normal and pathological classes using the Bhattacharyya coefficient was discussed. An unsupervised automatic training method is used for brain tumor segmentation[61].

In this study, an automated multi-modal diagnosis system that combines a support vector machine (ECOC-SVM) with an error-correcting output code and deep neural network to identify and locate the various kinds of brain tumors was proposed. This hybrid model served as both an extractor and a feature classifier. To locate the brain tumor inside the aberrant MRIs, a deep CNN including five layers of region-based (R-CNN) architecture was employed in the second step of the proposed approach. The proposed approach, which made use of an AlexNet, attained a high accuracy of 99.55%. Additionally, it outperformed alternative non-deep learning models[62].

This paper provided a brain tumor classification system based on features derived from a Gray Level Co-occurrence Matrix (GLCM) and convolutional neural network (CNN). For every image, they extracted four features (0°, 45°, 90°, and 135°) and fed these features into CNN. They tested their methodology on four different datasets (Mg-Gl, Mg-Pt, Gl-Pt, and Mg-Gl-Pt), with the Gl-Pt dataset yielding the best accuracy of 82.27% when they used two sets of features: contrast with homogeneity and contrast with correlation[63].

This research presented an automated approach for brain tumor detection and grading using deep neural networks. The method uses fuzzy C-Means (FCM) to segment the brain. Texture and shape features are then collected from these segmented regions and fed into SVM and DNN classifiers. The outcomes demonstrated that the accuracy rate attained by the system was 97.5%[64].

This study suggested a system that blends deep learning (DL) methods with discrete wavelet transform (DWT) characteristics. The brain tumor was segmented using the fuzzy c-mean method. Then, for each lesion that was found, the DWT was used to extract features. These features were then fed into principal component analysis (PCA) for feature dimension reduction, and finally, the features that were chosen were fed into deep neural networks (DNN). According to the data, they are able to attain a sensitivity of 97.0 % and an accuracy rate of 96.97%[65].

This research primarily addressed the usage of DWT brain and grey level co-occurrence matrix (GLCM) to reduce complexity and enhance performance. Here, PNNs are employed as classifiers and morphological techniques are applied to reduce noise. The experiment's final outcome was almost 100% accurate[66].

This study examined several MR image orientations and segmented the images using various networks. The experiment that was carried out yielded a die score of 0.79 in several networks and 0.73 in a single network[67].

Support vector machines (SVM) and self-organized mapping (SOM) were used in this work to identify the type of tumor and segment it from an MR image. The authors of this work were not able to identify and remove the appropriate tumor location from the picture. Owing to many lighting problems, the image contained extraneous white areas that could have been mistakenly identified as tumors. Additionally, the image's undesirable noise and decreased contrast highlight a number of areas that are mistakenly identified as tumors. The deteriorated quality of the MR image as a result of multiple issues that might have arisen during the acquisition stage was another difficulty[68].

This study uses a deep neural network correlation learning technique for CT brain tumor identification. To get the greatest possible detection result from the ANN, they adjusted them using CNN architecture palettes. The AISA framework for MRI data processing showed how to extract textural features and derive separate subspaces from brain scan data. Then, t-SNE embedding is used to reduce dimensionality for discriminative classification. The KNN classification is then used[69].

This research offered an extensive data augmentation strategy combined with CNN for brain tumor classification. The technique that uses segmented brain tumor MRI images to classify brain tumors into multiple grades. For classification utilizing transfer learning, they employed pretrained VGG-19 CNN architecture and obtained an overall accuracy of 87.38% and 90.67% for data prior to and during augmentation, respectively[70].

This study uses CNN in conjunction with neutrosophic expert maximum fuzzy (NS-CNN) sure entropy to classify brain tumors. After brain tumors were segmented using the neutrosophic set - expert maximum fuzzy-sure approach, the images were fed to CNN for feature extraction and subsequently to SVM classifiers for classification as benign or malignant. Their success rate on average was 95.62%[71].

This study presented a new CNN architecture for grading (classifying) brain malignancies into three classes (pituitary tumor, glioma, and meningioma) utilizing T1-weighted contrast-enhanced brain MR images. The suggested CNN classifier is an effective tool that performs well overall, achieving 98.93% accuracy and 98.18%

sensitivity for cropped lesions, 99% accuracy and 98.52% sensitivity for uncropped lesions, and 97.62% accuracy and 97.40% sensitivity for segmented lesion images[13].

In order to identify brain tumors, a stacked sparse auto encoder (SSAE) model with two fine-tuned layers hybridized with a high pass filter plus a median filter was employed in this study. The hyper-parameters were also adjusted after numerous experiments. Nevertheless, the employed hyper-parameter optimizer lacked speed and efficacy[72].

The researchers created a hybrid deep learning model that combined the best feature selection models (GLCM and GLRLM) with the Crow Search optimization technique to create an accurate classifier for identifying brain diseases and Alzheimer's disease. The suggested model outperformed CNN, NN, and SVM in terms of accuracy, sensitivity, and specificity, according to the accuracy data that were disclosed. However, the significance of the architecture's features and hyper-parameters was not evaluated[73].

Inception-v3 and DensNet201, two pre-trained CNN models, can help the deep learning model perform better in the categorization of brain tumor intensity[74]. This was demonstrated in this research. According to the results, both models outperformed tweaked VGG16 and CNN-ELM[75, 76].

In order to categorize brain cancers, this work proposed a computer-aided tumor detection model that consists of three CNN models (VGG16, GoogLeNet, and AlexNet) with an incorporated transfer learning and data augmentation technique. Up to 98.69% was the maximum accuracy that the VGG16 model suggested. However, VGG19 performed better than VGG16 in the majority of categorization case studies[75].

This research proposed a deep learning-based automated brain tumor detection model. The study is helpful in identifying the tumor under a microscope. This method designs a new 3D CNN model for the purpose of extracting the brain tumor. Next, feature extraction was trained using a CNN model that had already been trained. In the final stage, the best features were chosen, and experiment was conducted on BRATS 2015, 2017, and 2018. The percentage of accuracy on these three datasets is 92.67%, 96.97%, and 98.32%, respectively[77].

While more complicated nonlinear interactions can be extracted using deeper Convolutional Neural models, which can also decode and improve the accuracy of the model, most of them suffer from problems like vanishing or bursting gradients. The Residual Network (ResNet) modification is one of the best. The ability to train more intensive CNNs using shortcut connections to ignore one or more layers is a key advantage of ResNet models. Using MRI images as a basis, brain tumor segmentation and classification were one use case for the ResNet. ResNet, when used in conjunction with various augmentation techniques including rotation, shifting, and zooming, outperformed earlier research[78].

In this paper, CNN and depth-wise separable CNN algorithms are applied on the MR image dataset. Nevertheless, the technique can determine whether a tumor is there or not. But the algorithm was unable to identify a specific kind of tumor and recommend a course of treatment in accordance with that information[79].

This research presents the implementation and analysis of Artificial Neural Network (ANN) and Convolution Neural Network (CNN) for brain tumor detection. When using the testing data, the CNN-model produced an accuracy of 89%, whereas the ANN model produced an accuracy of 65.21%. More image data, nevertheless, can be provided to raise this. Additionally, they suggested using optimization techniques in their study to determine the maximum number of layers and filters that can be applied in a model to improve the system[80].

This paper proposed a hybrid system that included a convolutional neural network modified by the elephant herding optimization (EHO) method for the classification of glioma brain tumors. CNN's hyper-parameters were calibrated using the EHO, which also produced a classification improvement rate[81].

This paper proposed a novel automated deep learning technique for classifying brain tumors into multiple classes. The proposed approach is realized by fine-tuning and then training the Densenet201 Pre-Trained Deep Learning Model via a deep transfer of imbalanced data learning. The trained model's features are taken from the average pool layer, which has the extremely detailed information about every kind of tumor. Two feature selection strategies are proposed because the qualities of this layer are insufficient for a clear classification. Entropy-Kurtosis-based High Feature Values (EKbHFV) is the first technique and a modified genetic algorithm (MGA) based on metaheuristics is the second. The proposed new threshold function further refines the selected properties of the GA. Eventually, a multiclass SVM cubic classifier is utilized to classify the fused EKbHFV and MGA-based features utilizing a non-redundant serial-based method. Two datasets, BRATS2018 and BRATS2019, are employed in the experimental process without any increases and have produced results with an accuracy of over 95%. This work is significant as evidenced by the accurate comparison of the proposed approach with other neural nets[82].

An extended convolution neural network was proposed in this study to improve brain tumor classification performance using a tiny kernel and attain an accuracy rate of about 99% with a loss rate of 0.0033% leads to 0. Tensor Flow is used in the research project to categorize brain tumors through the use of magnetic resonance imaging, while Theano's Numpy array is utilized to handle complex expressions in Python. They impose an eight-layer convolutional neural network in Tensor Flow. The dataset included 1202 magnetic resonance imaging scans, 600 of which were classified as malignant and the rest as non-cancerous. 99.99% was the training accuracy, 99% was the validation accuracy, and across 15 epochs, the validation loss was 0.00333 to 0. The model was developed using both the GPU-based TFlearn and the CPU-based TensorFlow, which allows for substantially quicker model creation[83].

This study introduced an automated GoogLeNet architecture-based method for segmenting and detecting multi-sequence brain MRIs. According on the classification findings, GoogLeNet performed better than other machine learning techniques[84].

Using an MRI image dataset, the authors of this paper proposed an explanation-driven deep learning model (DL) for the prediction of discrete subtypes of brain tumors (meningioma, glioma, and pituitary) using a convolutional neural network (CNN), local interpretable model-agnostic explanation (LIME), and Shapley additive explanation (SHAP). In contrast to earlier models, the model included Gaussian noise to images of lower quality in order to overcome the classification challenge with metal artifacts and noise. Comparing the CNN training outcomes to other cutting-edge techniques, 94.64% of them show accuracy. Shapley values analyze all future predictions using all potential input combinations; hence SHAP was utilized to guarantee consistency and local correctness for interpretation. On the other hand, LIME builds sparse linear models surrounding every prediction to show the model's localized behavior[85].

2.2.2 Fixed Input Size Problem Associated with CNN

The image that we can obtain generally comes in different sizes. The development of convolution neural network (CNN)-based computer vision technologies has been made easier by the abundance of images. Target recognition, image categorization, and many other computer vision tasks have benefited greatly from CNN-based computer vision technology[86]. Currently, most CNNs are primarily made up of two parts: the convolution part and the fully-connected part that follows which makes most CNN models have a limit in terms of training and prediction. The fully-connected part requires uniformity in the size of all input data. A fixed size is not required in the convolutional section because the number of parameters is independent of the input size. It can create a feature map with a size proportionate to the input size given an input of any size. Conversely, the number of inputs and outputs of the neurons directly affects how many parameters the fully-connected layer has. The final output layer has a set number of neurons, which is equal to the number of classes. Iterative updates can only be used to learn parameters in networks where the number of parameters is fixed. The fully-connected layer means that a lot of CNNs need fixed-size input[87].

CNNs can attain state-of-the-art performance in many tasks and domains, but an intrinsic shortcoming of CNNs is their sensitivity to image size, which restricts practical use cases and necessitates matching evaluation inputs to training image size[88]. For instance, networks that are trained on a particular image size exhibit reduced performance when evaluated on other image sizes[89]. Moreover, preprocessed images with precisely the same width and height make up image recognition datasets because consistent input shape is a necessary requirement for most neural network architectures and a very useful condition for effectively training them. Consequently, depending on the input size specification of the CNN architecture, image recognition datasets are typically downsampled to a size of 256x256 or something similar. This extremely low resolution enables to fit all of the activations created in the typical CNN architectures in memory. In actuality, employing larger images creates new learning hurdles and problems with memory usage[90].

CNN architectures that accept smaller input images can use huge images in three different ways: by resizing the input image, by increasing the model size, or by processing images in batches. It is possible to adjust the input image to make it match the CNN's minimum input size requirement. Downsampling an input image to decrease its spatial resolution while maintaining its two-dimensional representation is one method of resizing it. But here's where the well-known aliasing issue arises: high-frequency variations like changing bright and dark colors will translate into low-frequency variations like constant light and dark colors. Cropping the image to the desired size is another way to minimize its size. In order to preserve the core content, which is typically more important and beneficial for the image, the image is typically cropped from the center[91].

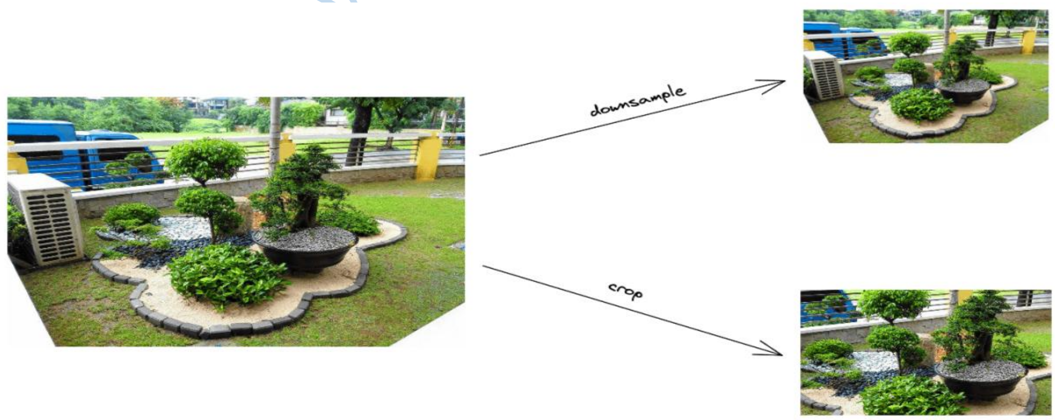


Figure 2.15: Resizing an Image Using Downsampling and Cropping[91]

Maintaining a constant input size and adjusting the CNN architecture appropriately is another way to use huge images in CNN designs that accept smaller input images. Convolutional layers in particular have the ability to shrink an input image's size.

Therefore, it is possible to create an architecture that can handle larger images by adding some more convolutional layers before the CNN model's convolutional layers. Reducing the total memory burden and processing the huge input images in chunks is the third way to solve this issue. More specifically, the inability of the entire training set to fit into memory occurs when employing huge images. Mini-Batch Gradient Descent provides the solution; in this method, we process a subset of the images each time we loop through the dataset. In order to accommodate the enormous images into the memory, we can therefore change the size of this group (batch) [91]. Batch training of images causes local features to be lost. Furthermore, labels must be created for each batch of images when training on image batches, which can take a lot of time[92].

In order to force all images into one consistent size when they have varying sizes, the standard technique is to crop or warp the image and then use CNN for training and learning. However, during the cropping procedure, a large number of pixels containing valuable information are lost, which will affect the training accuracy[86]. In numerous situations, cropping is not appropriate[87]. Geometric distortion results from warping distortion, which destroys structural information like an object's angle and proportion. This makes it challenging to learn about many scientific data sets with geometric labeling or high accuracy requirements. Warping might not provide too much of a hindrance in learning tasks for many natural picture datasets. However, as computer vision advances, CNN processes an increasing amount of scientific data. This implies that many application scenarios will be susceptible to warp, and the demand for warped image preprocessing will not be met[86].

A particular kind of CNN architecture known as fully-convolutional is described in the literature and enables the use of varied input forms. The architecture's primary feature is that it fixes its shape to enable fully-connected usage by aggregating all spatial information into a single value before feeding fully-connected layers. The ability to feed images of various forms might be resolved by the fully-convolutional layers, however learning patterns at such a large range of scales will be extremely challenging for CNN[90].

For the purpose of avoiding the fixed-size constraint in input, the fully-connected layer of the model can be substituted with Global Average Pooling (GAP), which was introduced in 2014[93]. The fully-connected layer continues to play a large and significant role in many computer vision applications, even if GAP has demonstrated good performance in certain workloads. Additionally, SPPNet, whose primary tactic is Space Pyramid Pooling, or SPP, was also proposed[94, 95]. The fixed-size input limit can be removed using this technique. SPP can generate an output with a constant size regardless of the dimensions or aspect ratio of the original image. SPPNet achieved significant success in maintaining the fully connected layer at the ILSVRC 2014[86, 96].

2.2.3 Input Size Limitation

A geometry image dataset sensitive to warp was constructed for this work in order to efficiently confirm if geometric distortion is produced by the model during preprocessing or training. Variable Step Pooling (VSP) and Variable Step Convolution (VSC) are two suggested SPP enhancements. With this technique, CNNs with fully-connected layers that can only process inputs of a given size may process inputs of a variable size and still learn well. The approach demonstrates robustness in the model when used with AlexNet and VGGNet. The solution may retain the accuracy of the original model on the public dataset VOC 2007 with unfixed image sizes and ImageNet with fixed image sizes, suggesting that the strategy is robust on common datasets that are not susceptible to geometric distortion[86].

A stochastic training regime, in which image sizes fluctuate with each optimization step, was proposed in this recent work. The study showed that models trained under the stochastic regime are more robust to changes in image size, enhanced model accuracy (generalization), and more resilient to fewer training loops[88].

Practitioners and earlier research have noted that a network trained on a particular input dimension can still be partially utilized at inference with a changed image size[97]. Additionally, accuracy can be improved up to a threshold by evaluating with an image size bigger than that used for training; after that, it rapidly deteriorates[89]. A recent study that employed a convolutional network to train and evaluate images revealed a trade-off between computational efficiency and accuracy[98]. This result is in line with previous research that showed training with a bigger image size can lead to a higher classification error[99, 100].

Furthermore, earlier research investigated the idea of progressive resizing, which entails enlarging images as training goes on to enhance model performance and convergence time [101, 102]. CNNs can be modified to a bigger size after training with a fixed small image size, which will be used for evaluation. At the expense of an additional fine-tuning step and additional computations at inference time, this procedure allowed for quicker training times and increased accuracy while reducing the train-test difference caused by the change in image size[89].

Before differentiating between classes, several pattern scales were handled using spatial pyramid pooling (SPP). In order to classify images based on their similar shapes, an input pipeline known as Buckets to Buckets (B2B) was also created. Based on the image of the museum piece, the major material was predicted using the Medium dataset, a subset of the recently made available Met dataset that includes a number of sculptures, paintings, and images. This dataset, which is appropriate for project, includes images with different aspect ratios and sizes[90].

This paper presents the Zoom-In network, an end-to-end CNN model that uses a single GPU to classify huge images with small objects by utilizing hierarchical attention sampling. Four large-image histology, road-scene, satellite, and gigapixel pathology datasets were used to assess the approach. According to experimental findings, the model uses less memory while achieving greater accuracy than current techniques[103].

It was shown that convolutional neural networks trained using streaming stochastic gradient descent (SSGD) could produce results comparable to those of convolutional neural networks that only retained a portion of the image in memory. The outcome of SSGD implementation demonstrates that it can potentially reduce network memory footprint and is numerically equivalent to regular stochastic gradient descent[92].

2.3 Summary of Gaps in Literature Reviewed

Different convolutional neural network models such as AlexNet VGGNet, GoogleNet, ResNet, DenseNet, and many more have been developed for the detection of brain tumors. Also, many researchers had retrained existing CNN models to reduce computation time and improve performance. The problem with almost all these existing networks is that they have fixed input size limitation. Majority of the researchers that have worked on brain tumor detection and classification using these CNN models adhere strictly to the specifications of the models. They warp and crop the brain tumor images to get the image fitted into the CNN models, classify the images and then evaluate the performance of the model. They concluded by saying their models perform better than existing models. The fixed input size limitation problem is not well covered in the literature. Only a small number of researchers have explored alternative approaches to using large images in CNN architectures that accept smaller input images.

In this study, the researcher developed a brain tumor-oriented CNN model that accepts varying input size. The model was trained with brain tumor MRI images of varying sizes collected from various repositories. To the best of the researchers' knowledge, no such CNN model has been developed for only an application area or an area in medical field. Presently, the researchers can only guarantee the effectiveness of this model on brain tumor MRI images. Its effectiveness on other image recognition datasets such as animals, human faces and the likes may be communicated in future research.

Endnotes

1. Brain & Spine Foundation. A guide for patients and carers. www.brainandspine.org.uk 2020, 1-56.

2. Cancer Council. *Understanding Brain Tumors: A Guide for People with Brain or Spinal Cord Tumors, their Families and Friends*. 2020. May 2020 edition, first published 1995, Next Ed. 2022.

3. Cancer Support Community. *Frankly Speaking about Brain Tumors.* www.cancersupportcommunity.org 2013, 1-96.

4. Open Access Publisher. *Brain Tumor*. 2022, https://www.omicsonline.org/open-access-journals-list.php.

5. K.A.A. Abolanle, S. Amina, A. Muhammad, A. Hina, & K.T. Omowumi. *Brain Tumor: An overview of the basic clinical manifestations and treatment.* **Glob J Cancer Ther.** 6(1), 2020, 038-041.

6. Q.T. Ostrom, & J.S. Barnholtz-Sloan. "*Current state of our knowledge on brain tumor Epidemiology".* Current Neurology and Neuroscience Reports 11: 2011, 329-335. Link: https://bit.ly/2T5lI7V.

7. M. K. Abd-ellah, A. I. Awad, A. M. Ashraf, & H. F. A. Hamed. *A Review on Brain Tumor Diagnosis from MRI Images: Practical Implications, Key Achievements, and Lessons Learned*. **Magnetic Resonance Imaging.** 61, 2019, 300–318.

8. N. Ghassemi, A. Shoeibi, & M. Rouhani. *Deep neural network with generative adversarial networks pre-training for brain tumor classification based on MR images*. **Biomedical Signal Processing and Control.** 57, 2020, 101678.

9. G. Litjens, T. Kooi, B.E. Bejnordi, A.A.A. Setio, F. Ciompi, M. Ghafoorian, J.A.W.M. Laak, B. Ginneken, & C.I. Sánchez. *A survey on deep learning in medical image analysis*. **Medical Image Analysis.** 42, 2017, 60–88.

10. Ö. F. Ertuğrul. *A novel clustering method built on random weight artificial neural networks and differential evolution*. **Soft Computing.** 24, 2020, 12067–12078.

11. L. Eyal, J. Rajan, R. Narges, M.F. Girish, & W.L. Yvonne. *State of the art: Machine learning applications in glioma imaging*. **American Journal of Roentgenology**, 212(1), 2019, 26-37.

12. M. Neha, K.M. Yogesh, M. Shruti, & M. Divya. *Detection of Brain Tumor in MRI Image through Fuzzy-Based Approach*. High-Resolution Neuroimaging-Basic Physical Principles and Clinical Applications, IntechOpen, 2018.

13. M.A. Ali, A. Hiam, A.Q. Isam, A. Amin, & A. Wafaa. *Brain Tumor Classification Using Deep Learning Technique - A Comparison between Cropped, Uncropped, and Segmented Lesion Images with Different Sizes*. **International Journal of Advanced Trends in Computer Science and Engineering.** 8(6),

2019, 3684– 3691.

14. S. Wojciech, W. Thomas, & M.K. Robert *Explainable Artificial Intelligence: Understanding, Visualizing and Interpreting Deep Learning Models*. **arXiv:1708.08296 [cs.AI],** 2017.

15. L, Cao. *Data Science: A Comprehensive Overview*. **ACM ComputSurv (CSUR)**. 50(3), 2017, 43.

16. I.H. Sarker, M.M. Hoque, U. MdK, & A. Tawfeeq. *Mobile Data Science and Intelligent Apps: Concepts, AI-Based Modeling and Research Directions*. **Mob Netw Appl.** 2020, 1–19.

17. I.H. Sarker. *Machine Learning: Algorithms, Real World Applications and Research Directions*. **A SPRINGER NATURE Journal. SN Computer Science.** 2(160), 2021, 1-21.

18. Amapiano. *Introduction to Machine Learning 2023: PDF Download. A-Brief-Introduction-to-Machine-Learning-for-Engineers*. 2023, https://www.citimuzik.com/2023/03/introduction-to-machine-learning-2023-pdf-download.html.

19. K.P. Anubhav, K.M. Dileep, D.J.M. Víctor, B.R. Mohan, & P. Achyutha. *Machine Learning Approach for Prediction of the Online User Intention for a Product Purchase*. **International Journal on Recent and Innovation Trends in Computing and Communication,** 11(1s), 2023, 41-51.

20. M.M. Vidya. *Covid-19 Prediction using Machine Learning*. **International Research Journal of Modernization in Engineering Technology and Science**. 05(05), 2023. 1180-1185.

21. M. Mohammed, M.B. Khan, & M.B.E Bashier. *Machine Learning: Algorithms and Applications.* **CRC Press.** 2016.

22. J. Han, J. Pei, & M. Kamber. *Data Mining: Concepts and Techniques*. **Amsterdam: Elsevier.** 2011.

23. I.H. Witten, & E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques.* A volume in the Morgan Kaufmann Series in Data Management Systems. Third Edition, 2011, 607-629.

24. 24. I.H. Sarker. *Deep Cybersecurity: A Comprehensive Overview from Neural Network and Deep Learning Perspective*. **SN Comput Sci**. 2021.

25. I.H. Sarker, P. Watters, & A.S.M. Kayes. *Effectiveness Analysis of Machine Learning Classification Models for Predicting Personalized Context-Aware Smartphone Usage*. **J Big Data**. 6(1), 2019, 1–28.

26. I.H. Sarker. *Machine Learning: Algorithms, Real World Applications and Research Directions*. **A SPRINGER NATURE Journal. SN Computer Science.** 2(160), 2021, 1-21.

27. McCallum A. *Information Extraction: Distilling Structured Data from Unstructured Text*. **Queue**. 3(9), 2005, 48–57.

28. Rupam. *What are Machine Learning Applications? Top 10 Industry and Real-World Use Cases.* 2022, https://emeritus.org/blog/machine-learning-what-are-machine-learning-applications/.

29. B. Jason. *What is Deep Learning?* 2020, https://machinelearningmastery.com/what-is-deep-learning/.

30. Y. LeCun, Y. Bengio, & G. Hinton. *Deep Learning*. **Nature**. 521(7553), 2015, 436–44.

31. Z. Zhang, P. Cui, & W. Zhu. *Deep Learning on Graphs: A Survey*. **IEEE Trans Knowl Data Eng.** 29813, 33, 2020.

32. A. Shrestha, & A. Mahmood. *Review Of Deep Learning Algorithms And Architectures*. **IEEE Access**.7:53040–65, 2019.

33. A. Laith, Z. Jinglan, J.H. Amjad, A. Ayad, D. Ye, A. Omran, J. Santamaría, A.F. Mohammed, A. Muthana, & F. Laith. *Review of Deep Learning: Concepts, CNN Architectures, Challenges, Applications, Future Directions*. **Journal of Big Data**. 2021, 1–74.

34. Coursera. *What Is Deep Learning? Definition, Examples, and Careers*. 2023, https://www.coursera.org/articles/what-is-deep-learning.

35. Techjury. *How Much Data is Created Every Day in 2022?* 2022, https://techjury.net/blog/how-much-data-is-created-every-day/#gref.

36. Saumyasaxena2730. *Introduction to Deep Learning*. 2023, https://www.geeksforgeeks.org/introduction-deep-learning/.

37. Y.N. Chen, K.C. Fan, Y.L. Chang, & T. Moriyama. *Special Issue Review: Artificial Intelligence and Machine Learning Applications in Remote Sensing*. **Remote Sens.** 15(569), 2023, Page 1-10.

38. O. Artem. *What Is Deep Learning and How Does It Work?* 2022, https://builtin.com/authors/artem-oppermann.

39. A.E. Maxwell, T.A. Warner, & L.A. Guillen. *Accuracy Assessment in Convolutional Neural Network-Based Deep Learning Remote Sensing Studies—Part 1: Literature Review.* **Remote Sens.** 13, **2021**, 2450.

40. Y. Wang, G.Y. Wei, & D. Brooks. *Benchmarking TPU, GPU, and CPU Platforms for Deep Learning*. **arXiv.** arXiv:1907.10701. 2019**.**

41. MathWorks. *What Is Deep Learning? Apply deep learning to the design of smart engineering systems*. 2023, https://www.mathworks.com/discovery/deep-learning.html.

42. M. Volodymyr, K. Koray, S. David, A.R. Andrei, V. Joel, G.B. Marc, G. Alex, R. Martin, K.F. Andreas, O. Georg, P. Stig, B. Charles, S. Amir, A. Ioannis, K. Helen, K. Dharshan, W. Daan, L. Shane, & H. Demis. *Human-Level Control Through Deep Reinforcement Learning*. **Nature**. 518, 2015. 529–533.

43. S.C. Nagesh. *Optimization Algorithms in Neural Networks*. KDnuggets. 2020, https://www.kdnuggets.com/2020/12/optimization-algorithms-neural-networks.html.

44. D. Sanket. *Various Optimization Algorithms for Training Neural Network. Towards Data Science.* 2019, https://towardsdatascience.com/optimizers-for-training-neural-network-59450d71caf6.

45. G. Ayush. *A Comprehensive Guide on Optimizers in Deep Learning.* Analytics Vidhya. 2023, https://www.analyticsvidhya.com/blog/2021/10/a-comprehensive-guide-on-deep-learning-optimizers/.

46. M. Faisal, A. Shabir, & K.W. Taeg. *An Efficient Optimization Technique for Training Deep Neural Networks.* **Mathematics.** 11(6), 2023, 1360.

47. S.W. Anish. *Types of Optimization Algorithms used in Neural Networks and Ways to Optimize Gradient Descent.* Nerd for Tech. 2021, https://medium.com/nerd-for-tech/types-of-optimization-algorithms-used-in-neural-networks-and-ways-to-optimize-gradient-descent-1e32cdcbcf6c.

48. K. Ajitesh. *Different Types of CNN Architectures Explained: Examples*. Analytics Yogi. 2023, https://vitalflux.com/different-types-of-cnn-architectures-explained-examples/.

49. M. Derrick. *Image Classification with Convolutional Neural Networks (CNNs).* KDnuggets. 2022, https://www.kdnuggets.com/2022/05/image-classification-convolutional-neural-networks-cnns.html.

50. B. Chinmay. *A Practical Guide to Selecting CNN Architectures for Computer Vision Applications from LeNet to EfficientNet: Choosing the Best CNN Architecture for Your Project.* 2023, https://levelup.gitconnected.com/a-practical-guide-to-selecting-cnn-architectures-for-computer-vision-applications-4a07ef90234.

51. H. Zubair. Best CNN Architecture for Image Processing. 2022, https://www.folio3.ai/blog/best-cnn-architecture-for-image-processing/.

52. B. Ed, & B. Kate. *Definition deep learning*. Techtarget. 2021, https://www.techtarget.com/searchenterpriseai/definition/deep-learning-deep-neural-network.

53. I. Ali, D. Cem, & Ş. Melike. *Review of MRI-based Brain Tumor Image Segmentation Using Deep Learning Methods*. **Procedia Computer Science.** 102 (2016), 2016, 317 – 324.

54. M. S. Wessam, & S. Ahmed. *A Novel Framework for Brain Tumor Detection Based on Convolutional Variational Generative Models*. **Multimedia Tools and Applications**, 81, 2022, 16441–16454.

55. S. Khawaldeh, U. Pervaiz, A. Rafiq, & R.S. Alkhawaldeh. *Noninvasive Grading of Glioma Tumor Using Magnetic Resonance Imaging with Convolutional Neural Networks.* **Applied Sciences**, 8(1), 2018, 27.

56. A. Athency, B.M.A. Ancy, K.A. Fathima, R. Dilin, & M.C. Binish. *"Brain Tumor Detection and Classification in MRI"* **International Journal of Innovative Research in Science, Engineering and Technology**. 6(5), 32, 2017.

57. J. S. Paul, A. J. Plassard, B. A. Landman, & D. Fabbri. *Deep Learning for Brain Tumor Classification*. Proceedings of the SPIE, Volume 10137, id. 1013710 16 pp. 2017.

58. X.W. Gao, R. Hui, & Z. Tian. *Classification of CT Brain Images Based on Deep Learning Networks.* **Computer Methods and Programs in Biomedicine**, *138*, 2017, 49–56.

59. H. Shamsul, Y. John, F.J. Herbert, M.H. Mohammad, F. Giancarlo, & B. Michael. *A Hybrid Feature Selection with Ensemble Classification for Imbalanced Healthcare Data: A Case Study for Brain Tumor Diagnosis.* ***IEEE Access,*** **4**: 2017, 9145 - 9154.

60. C. Lina, & L. Shahram. Department of Electrical and Computer Engineering University of Nevada, Las Vegas (UNLV) *Machine Learning and Deep Learning Techniques to Predict Overall Survival of Brain Tumor Patients using MRI.* 2017 IEEE 17th International Conference on Bioinformatics and Bioengineering.

61. P. Chinmayi, L. Akilandeswari, M. Prabu Kumar, & K. Murali Babu. *An Efficient Deep Learning Neural Network Based Brain Tumor Detection System*" **International Journal of Pure and Applied Mathematics.** 117(17), 2017, 151-160.

62. M.K. Abd-Ellah, A.I. Awad, A.A.M Khalaf, & A., H. F. *Two-Phase Multi-Model Automatic Brain Tumour Diagnosis System from Magnetic Resonance Images Using Convolutional Neural Networks*. **EURASIP Journal on Image and Video Processing**, 97, 2018.

63. W. Widhiarso, Y. Yohannes, & P. Cendy. *Brain Tumor Classification Using Gray Level Co-occurrence Matrix and Convolutional Neural Network.* **IJEIS (Indonesian Journal of Electronics and Instrumentation Systems).** 1 8, 2018, 179-190.

64. J. Seetha, & S.S. Raja. *Brain Tumor Classification Using Convolutional Neural Networks*. **Biomed. & Pharmacol. J.** 11(3), 2018, 1457-1461.

65. M. Heba, A.E. El-Sayed, M. E. El-Sayed, & M.S. Abdel-Badeeh. *Classification Using Deep Learning Neural Networks for Brain Tumors*" **Future Computing**

and Informatics Journal. 3, 2018, 68-71.

66. N. V. Shree, & T. N. R. Kumar. *Identification and Classification of Brain Tumor MRI with Feature Extraction Using DWT and Probabilistic Neural Network.* **Brain Infomatic.** 5(1), 2018, 23-30.

67. Zahra, S. Safiyeh, R. Alireza, N. Mehdi, A. Hamidreza, Z. Nader, K. Ali, E. & Shadrokh, S. *Brain Tumor Segmentation Using Deep Learning by Type Specific Sorting of Images*. International conference on Computer Vision and Pattern Recognition (CVPR)*,* 20th Sep 2018.

68. K.Saurabh, A. Iram, G. Shubhi, K.S. Anand, & J. Vivek. *Brain Tumor Detection Using Image Processing*. **International Journal of Information Sciences and Application (IJISA).** 11(1), 2019, 20 – 25.

69. Q. Ke, J. Zhang, W. Wei, R. Damasevicius, & M. Wozniak. *Adaptive Independent Subspace Analysis of Brain Magnetic Resonance Imaging Data*. **IEEE Access**. 7, 2019, 12252–12261.

70. M. Sajjad, S. Khan, K. Muhammad, W. Wu, A. Ullah, & S.W. Baik. *Multi-Grade Brain Tumor Classification Using Deep CNN with Extensive Data Augmentation*. **Journal of Computational Science.** 30, 2019, 174-82.

71. F. Özyurt, E. Sert, E. Avci, & E. Dogantekin. *Brain Tumor Detection Based on Convolutional Neural Network with Neutrosophic Expert Maximum Fuzzy Sure Entropy*. **Measurement**, 147, 2019.

72. J. Amin, M. Sharif, N. Gul, M. Raza, M.A. Anjum, M.W. Nisar, & S.A.C Bukhari. *Brain Tumor Detection by Using Stacked Autoencoders in Deep Learning*. **Journal of Medical Systems**. 44(2), 2020.

73. 73 R.J.S. Raj, S. J. Shobana, I.V. Pustokhina, D.A. Pustokhin, D. Gupta, & K. Shankar. *Optimal Feature Selection-Based Medical Image Classification Using Deep Learning Model in Internet of Medical Things*. **IEEE Access**, *8*, 2020, 58006–58017.

74. N. Noreen, S. Palaniappan, A. Qayyum, I. Ahmad, & M.I.M. Shoaib. *A Deep Learning Model Based on Concatenation Approach for the Diagnosis of Brain Tumor*. **IEEE Access.** 8, 2020, 55135–55144.

75. A. Rehman, S. Naz, M.I. Razzak, F. Akram, & M. Imran. *A Deep Learning-Based Framework for Automatic Brain Tumors Classification Using Transfer Learning*. **Circuits, Systems, and Signal Processing.** 39, 2020. 757–775.

76. A. Pashaei, H. Sajedi, & N. Jazayeri. *Brain Tumor Classification via Convolutional Neural Network and Extreme Learning Machines*. 2018 8th International Conference on Computer and Knowledge Engineering (ICCKE), 2018 314–319.

77. A. Rehman, M. A. Khan, T. Saba, Z. Mehmood, U. Tariq, & N. Ayesha. *Microscopic brain tumor detection and classification using 3D CNN and feature selection architecture*. **Microsc Res Tech**, 84(1), 2020, 133–149.

78. S. A. A. Ismael, A. Mohammed, & H. Hefny. *An Enhanced Deep Learning Approach for Brain Cancer MRI Images Classification Using Residual Networks.* **Artificial Intelligence in Medicine**, 102, 2020, 101779.

79. B. Kavita, R. Varun, S. Sanjay, & S. Vijay. *Brain Tumor Detection Using Deep Learning Techniques.* Proceedings of the 4th International Conference on Advances in Science & Technology (ICAST2021), 2021, 1-5.

80. P. Gokila Brindha, M. Kavinraj, P. Manivasakam, & P. Prasanth. *Brain Tumor Detection from MRI Images Using Deep Learning Techniques.* 2021 IOP Conference Series: Materials Science and Engineering. 1055, 2021, 012115.

81. T. Bezdan, S.K.V. Milosevic, M. Zivkovic, N. Bacanin, & I. Strumberger. *Optimizing Convolutional Neural Network by Hybridized Elephant Herding Optimization Algorithm for Magnetic Resonance Image Classification of Glioma Brain Tumor Grade*. Zooming Innovation in Consumer Technologies Conference (ZINC), 2021, 171–176.

82. I. S. Muhammad, A. K. Muhammad, A. Musaed, A. Khursheed, & R. Mudassar. *A decision support system for multimodal brain tumor classification using deep learning.* **Complex & Intelligent Systems**, 2021, 1-14.

83. H.U. Rehman, S. Masood, & M.A. Khan. *Brain Tumor Classification using Deep Learning Methods.* **International Journal of Information Systems and Computer Technologies (IJISCT).** Vol. 1(1), 2022, 37-53.

84. F. Ehret, D. Kaul, H. Clusmann, D. Delev, & J.M. Kernbach. *Machine Learning-Based Radiomics in Neuro-Oncology*. **Machine Learning in Clinical Neuroscience,** 134, 2022, 139–151.

85. L. Gaur, M. Bhandari, T. Razdan, S. Mallik, & Z. Zhao. *Explanation-Driven Deep Learning Model for Prediction of Brain Tumour Status Using MRI Image Data*. **Front. Genet.** 13:822666. 2022.

86. Q. Yili, K. Yaobin, & Y. Wei. *A Solution for Input Limit in CNN Due to Fully Connected Layer.* **IEEE Xplore.** 2018, 1-6.

87. K. He, X. Zhang, S. Ren, & J. Sun. *Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition.* **[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence**, 37(9), 2015, 1904-1916.

88. H. Elad, W. Berry, H. Itay, B. Tal, H. Torsten, & S. Daniel. *Mix & Match: Training Convnets with Mixed Image Sizes for Improved Accuracy, Speed and Scale Resiliency*. **arXiv.** 2019, 1-13.

89. T. Hugo, V. Andrea, D. Matthijs, & J. Herv´e. *Fixing the train-test resolution discrepancy*. **CoRR**, abs/1906.06423, 2019.

90. P. Ferran, G. Dario, & L. Jesus. *Training CNNs using high-resolution images of*

variable shape. **6th BSC Severo Ochoa Doctoral Symposium.** 2023, 60-61.

91. A. Panagiotis. *How to Handle Large Images to Train CNNs?* 2023, https://www.baeldung.com/cs/.

92. J.H.F.M. Pinckaers, & G.J.S. Litjens. *Training Convolutional Neural Networks with Megapixel Images*. 1st international conference on Medical Imaging with Deep Learning (MIDL 2018), Amsterdam. 2018, 1-3.

93. M. Lin, Q. Chen & S. Yan. Network in Network. **Computer Science**, 2013.

94. K. Grauman & T. Darre ll. *The Pyramid Match Kernel: Discriminative Classification with Sets of Image Features*, in ICCV, 2005.

95. S. Lazebnik, C. Schmid, & J. Ponce. *Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories* in **CVPR**, 2006.

96. O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A.C. Berg, L. Fei-Fei. *Imagenet Large Scale Visual Recognition Challenge.* **arXiv:1409.0575 [cs.CV]**, 3, 2015, 1-43.

97. S. Karen, & Z. Andrew. *Very Deep Convolutional Networks for Large-Scale Image Recognition.* **arXiv preprint** arXiv:1409.1556, 2014.

98. T. Mingxing, & L. Quoc. *Efficientnet*: *Rethinking Model Scaling for Convolutional Neural Networks*. In International Conference on Machine Learning, 2019, 6105–6114.

99. S. Christian, V. Vincent, I. Sergey, S. Jon, & W. Zbigniew. *Rethinking the Inception Architecture for Computer Vision*. In Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, 2818–2826.

100. H. Yanping, C. Yonglong, C. Dehao, L. HyoukJoong, N. Jiquan, V.L. Quoc, & C. Zhifeng. *Gpipe: Efficient Training of Giant Neural Networks Using Pipeline Parallelism*. **arXiv preprint** arXiv:1811.06965, 2018.

101. K. Tero, A. Timo, L. Samuli, & L. Jaakko. *Progressive Growing of Gans for Improved Quality, Stability, and Variation*. **arXiv preprint** arXiv:1710.10196, 2017.

102. H. F. Jeremy. *Progressive Resizing.* 2018, https://www.fast.ai/2018/04/30/dawnbench-fastai/.

103. K. Fanjie, & H. Ricardo. *Efficient Classification of Very Large Images with Tiny Objects*. **Computer Vision Foundation (CVF) Open Access**. 2022, 2384-2394.

104. L. Gaur, M. Bhandari, T. Razdan, S. Mallik, & Z. Zhao. *Explanation-Driven Deep Learning Model for Prediction of Brain Tumour Status Using MRI Image Data*. **Front. Genet.** 13:822666. 2022.

Chapter Three

Methodology

3.1 Research Approach

A research method is the process used to do a specific type of study. It refers to the methods, procedures, or approaches used in the gathering of information or proof for analysis with the goal of learning new things or developing a deeper comprehension of a subject. Developmental research design was chosen as the research methodology in order to meet the study's objectives and address the research issues. Furthermore, the nature of the research is quantitative. To be more precise, developmental research is the methodical investigation of design, development, and assessment procedures with the goal of providing an empirical foundation for the development of tools and products, both instructional and non-instructional, as well as new or improved models that direct their evolution[1]. It is also known as the methodical study of creating, refining, and assessing educational procedures, products, and programs that have to adhere to strict internal consistency and effectiveness standards[2].

The first and most popular kind of developmental research focuses on scenarios where developmental process of a product is examined and explained and the end product is assessed. Another kind of developmental study is primarily concerned with how the product affects the organization or the learner. A general examination of the design creation or evaluation processes, either as a whole or as individual components, is the focus of a third kind of developmental research[2].

Practical application is the focus area of developmental study. Research projects are typically grounded in a tangible problem or intricate matter pertaining to education, learning, or development within a particular setting. It focuses on creating, carrying out, and assessing a tangible solution that will address the issue or challenge that has been recognized. The foundation of developmental research can also be the identification of a certain need for modification, improvement, or enhancement, such as the development of new digital technology-based learning opportunities[3]. Additionally, this study is quantitative in nature. When it comes to the design, development, deployment, and upkeep of Information Technology systems and services, quantitative research can be a useful tool[4]. Researchers can assess the model's dependability and performance using quantitative research approaches. By using the data gotten from quantitative research, problems can be found and fixed, and system performance can as well be improved[4]. Thus, the objectives of this developmental and quantitative research are to: (1) develop a brain tumor-oriented CNN model that will accept varying input sizes of brain tumor images (2) classify brain tumor MR brain tumor images and (3) evaluate the model to see how well it performs.

3.1.1 Area of Study

The area of study for this research is image classification, specifically brain tumor MR image classification. Assigning one or more labels to a complete image based on previously collected training data of previously labeled images is the process of image classification. Image classification is sometimes referred to as image recognition. To prevent data inconsistencies throughout the training phase, data labeling must be done precisely. During the model training phase, publically available datasets are frequently employed to enable correct data labeling[5].

Images are analyzed in a computer in the form of pixels. It accomplishes this by viewing the image as an array of matrices, where the resolution of the image determines the size of the matrix. From a computer's perspective, image classification is the application of algorithms to analyze this statistical data. Image classification in digital image processing is achieved by automatically classifying pixels into predefined groups, or so-called classes[6]. The techniques reduce the workload on the final classifier by segmenting the image into a set of its most salient features. These features provide the classifier with an understanding of the image's meaning and potential classification. Since the other steps in the classification process rely on the feature extraction process, it is therefore the most crucial step in the process. The data that is provided to the algorithm plays a major role in image classification, especially in the case of supervised classification. A poorly optimized dataset with uneven data based on class, low-quality images, and poorly annotated images performs poorly compared to a well-optimized dataset[6].

The process of classifying images involves image pre-processing, object detection (which involves segmenting the image to find the object's location within the set of photos), object recognition, and object training (labeling the located images and using Artificial Intelligence models to train the label image) and finally, object classification, which is the concluding stage of the procedure, when the model is prepared to classify the images[7]. Applications for image classification are found in various domains, including machine vision, traffic control systems, medical imaging, item recognition in satellite images, and so on[6]. The kind of image classification methodology to be used will depend on the nature of the problem. They are binary classification, multiclass classification, multilabel classification, and hierarchical classification.

Binary Classification: Binary classification utilizes an either-or logic to label images and classifiy unknown data points into two groups. Binary classification is used to handle many different yes/no problems, such as classifying benign or malignant tumors, analyzing product quality to determine whether a product has faults, and many more tasks requiring judgment calls.

Multiclass Classification: Multiclass classification, as the name implies, group objects into three or more classes, whereas binary classification is used to discriminate between two classes of objects. It's incredibly helpful in many fields, such as Natural Language Processing, where multiple emotions are present, like sentiment analysis. Additionally, it is employed in medical diagnostics, such as when separating illnesses into several groups.

Multilabel Classification: Multilabel classification permits the object to be allocated to numerous labels, in contrast to multiclass classification, which assigns each image to a single class. For instance, there can be multiple colors in an image that needs to be classified. Consequently, several colors will be used as labels on a single image.

Hierarchical Classification: The task of classifying classes into a hierarchical structure based on their resemblance is known as hierarchical classification. A lower-level class is more definite and detailed, while a higher-level class represents larger categories[5].

3.2 Requirements Specification

The modified AlexNet CNN model was developed using both hardware and software tools from implementation to evaluation.

3.2.1 Hardware Specification

The training and testing of the modified AlexNet CNN model were performed on Intel(R) Core (TM) i5-1035G1 CPU @ 1.00GHz 1.19GHz with 16.0 GB RAM.

3.2.2 Software Specification

3.2.2.1 Integrated Development Environment

An integrated development environment (IDE) is the collection of applications that assist computer programmers create software code efficiently. It maximizes the productivity of software developers by combining skills such as software creation, editing, testing, and packaging in a user friendly application[10]. The integrated development environment (IDE) that was used to develop this brain tumor-oriented CNN model is Jupyter notebook. With the help of Jupyter Notebook, users may create and share interactive notebook documents that include text, videos, data visualizations, live code, and other computational outputs.

The program, which was developed by Project Jupyter, is open-source software and it is compatible with more than 40 computer languages, including R, Python, and Scala. Jupyter Notebook may run cells in any order and displays real-time code results and images. For speedy code experimentation, creating code presentations, or streamlining data science operations, Jupyter Notebook is a helpful tool. Jupyter Notebook is useful for developing initial code. It enables code segmentation and reruns of sections while retaining the values of variables from previously executed segments. Because of its interactive features and capacity to integrate code and visualization into a single document, Jupyter Notebook is widely used in data research and scientific computing. It also makes document sharing and collaboration simple[9].

3.2.2.2 Python

Python, a widely used object-oriented programming language, was created by Guido Rossum in 1989. It is perfect for quickly prototyping intricate applications. It is extendable to C or C++ and offers interfaces to numerous OS system calls and libraries. Python is a programming language used by many major corporations, such as NASA, Google, YouTube, BitTorrent, etc. Neural networks, artificial intelligence, natural language generation, and other cutting-edge computer science domains all make extensive use of Python programming. The Python programming language is useful for creating algorithms for artificial intelligence as well as for a variety of scientific applications, including statistical models[12].

3.2.2.3 TensorFlow

TensorFlow is an open-source machine learning framework created by Google. It is one of the most popular platforms for creating and implementing machine learning models, such as convolutional neural networks. For creating and implementing models, TensorFlow offers an extensive array of tools and libraries, including low-level APIs like TensorFlow Core and high-level APIs like Keras, which enable more freedom and control over the model architecture. Large models can be trained concurrently on numerous machines by utilizing the distributed training features included in TensorFlow. TensorFlow's compatibility with a broad range of hardware, such as CPUs, GPUs, and TPUs, is one of its main advantages. This makes it possible to compute and train models efficiently, which is beneficial for tasks requiring a lot of data and processing. TensorFlow incorporates tools for model deployment and optimization as well, which can enhance the effectiveness and performance of models[13].

3.2.2.4 NumPy

Numerical Python, or NumPy, is a general-purpose library for array processing[14]. A multidimensional array object, different derived objects (like masked arrays and matrices), and a variety of routines for quick array operations—like sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation, and much more—are all provided by this Python library[15]. In addition to its apparent applications in science, NumPy in Python can be utilized as a productive multi-dimensional data container. Numpy enables for the definition of any data types, which facilitates the quick and easy integration of NumPy with a wide range of databases[14].

3.2.2.5 Python Image Processing (PIL)

The practice of programmatically altering .jpg, .jpeg, .png, .tiff, .webp, .gif or any other type of image file is known as image processing[16]. With the help of image processing, it is possible to alter and modify thousands of images at once and gain insightful knowledge from them. One of the most popular programming languages for this is Python. Its incredible frameworks and tools make image processing a highly efficient task[17].

3.2.2.6 OpenCV2

An image processing tool used in computer vision and machine learning is called Open-Source Computer Vision Library (OpenCV). More than 2000 optimized algorithms for computer vision and machine learning tasks are included in this library[17]. OpenCV is designed using C++ with bindings for Python and Java[16]. It can process videos and images to recognize objects, faces, or even human handwriting[18].

3.3 Research Design

The research design phase decides how the model will operate in terms of data collection, data preprocessing, model building, model training, model testing and model evaluation. It also includes the hardware and software that will be used in designing the model. In other word the steps in the design phase determine exactly how the model will operate.

3.3.1 Dataset

In image classification, particularly supervised classification, the data fed into the algorithm is essential. The accuracy of the model increases with the quality of the data. An excellent training dataset improves the accuracy and dependability of the AI model's predictions and permits more thoughtful decision-making[7]. Brain tumor MRI images are the kind of data required for this research project. The brain tumor MRI dataset was collected from public domains. The brain tumor MRI images dataset is a significant resource for constructing and assessing machine learning models, notably Convolutional Neural Networks (CNNs), for the automated detection and categorization of brain tumors[19].

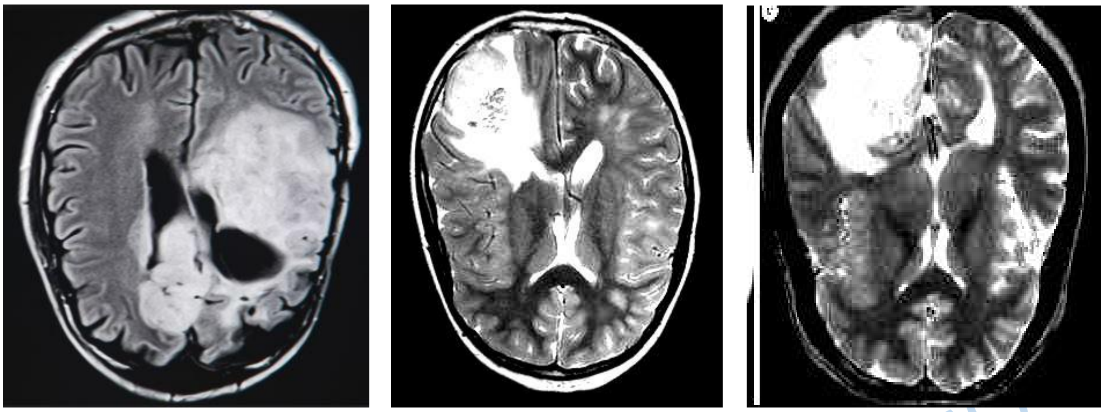
3.3.1.1 Data Collection Method

The brain tumor MRI dataset needed for this study was collected from Kaggle. The link is https://kaggle.com/. Kaggle has many menus which include Competitions, Datasets, Models, Code, Discussions, and Courses. The dataset module was selected and the researcher search for brain tumor images. Four hundred and twenty-nine (429) brain tumor dataset were available as at March 2, 2024. The under listed dataset were some of those selected from the available four hundred and twenty-nine (429) brain tumor dataset.

Table 3.1: Kaggle Brain Tumor Dataset

S/N	Brain Tumor Dataset	Link
1	Br35h :: brain tumor detection 2020	https://www.kaggle.com/datasets/ahmedhamada0/brain-tumor-detection
2	Br35h :: brain tumor detection 2020	https://www.kaggle.com/datasets/mikhaelapg/brain-tumor-detection
3	Brain tumor MRI dataset	https://www.kaggle.com/datasets/masoudnickparvar/brain-tumor-mri-dataset
4	Brain MRI images for brain tumor detection	https://www.kaggle.com/datasets/navoneel/brain-mri-images-for-brain-tumor-detection?select=yes
5	Crystal clean: brain tumors MRI dataset	https://www.kaggle.com/datasets/mohammadhossein77/brain-tumors-dataset
6	Brain_tumor_classification	https://www.kaggle.com/datasets/prathamgrover/brain-tumor-classification
7	Brain MRI scans for brain tumor classification	https://www.kaggle.com/datasets/shreyag1103/brain-mri-scans-for-brain-tumor-classification
8	Brain MRI images for brain tumor detection	https://www.kaggle.com/datasets/jjprotube/brain-mri-images-for-brain-tumor-detection8

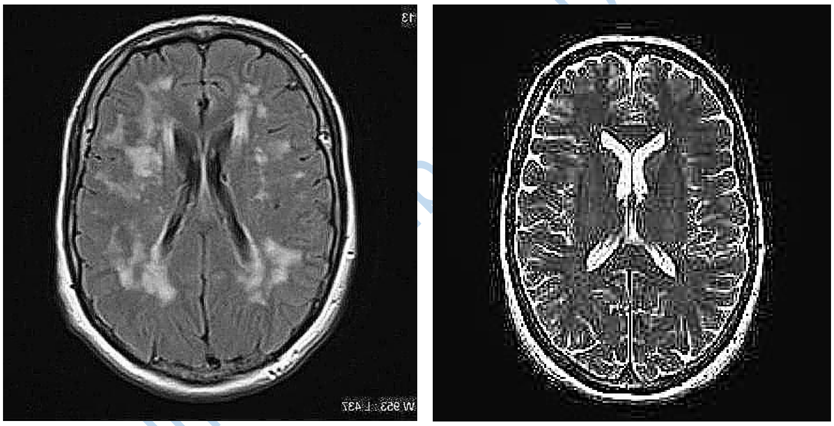
Researcher’s Concept, 2024



287 x 348 pixels 189 x 173 pixels 313 x 360 pixels

Figure 3.1: Brain Tumor Images and Their Varying Sizes Collected from Kaggle

Researcher’s Concept, 2024



441 x442 pixels 225 x 225 pixels

Figure 3.2: Normal Brain Images and Their Varying Sizes Collected from Kaggle

Researcher’s Concept, 2024

The total number of cancerous images collected from Kaggle was four thousand three hundred and four (4304) and non-cancerous images was four thousand one hundred and sixty-five (4165) making a total of eight thousand four hundred and sixty-nine (8469). All the images were of varying sizes.

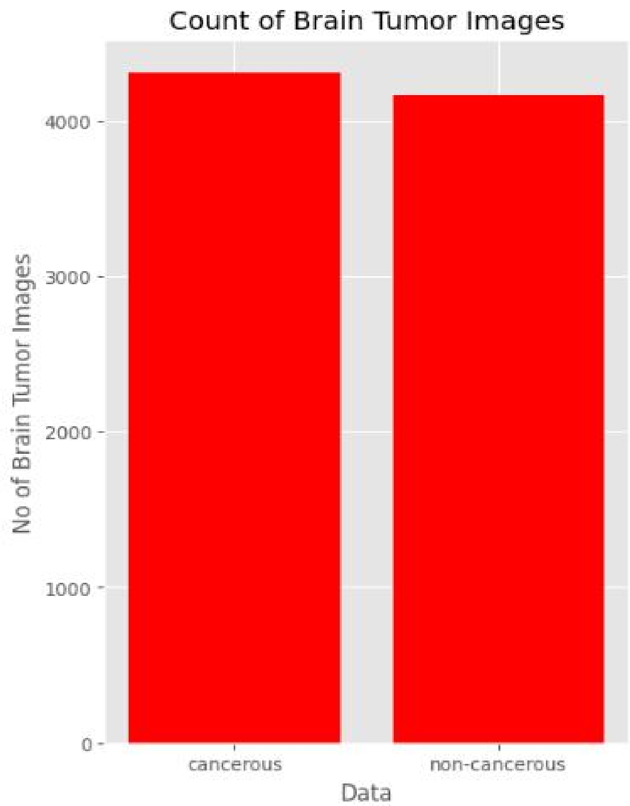


Figure 3.3: Histogram Showing the Number of Cancerous and Non-Cancerous Brain Tumor Images

Researcher's Concept, 2024

3.3.2 Image Preprocessing

The process of applying techniques to enhance the quality of the image data and get it ready for the next steps is known as image preprocessing. Among the tasks that can be performed during the picture processing stage are data augmentation, data cleansing, noise reduction, image resizing, cropping, and normalization[5]. The following were the main pre-processing techniques applied to the brain tumor MR image:

3.3.2.1 Data Cleansing

When preparing data for CNN model training, data cleansing techniques like removing duplicate data, deleting irrelevant data, filtering outliers, finding missing data, and fixing structural errors are crucial because inaccurate data can result in inaccurate image classification models[7]. Before creating this model, all unnecessary and undesired images were extracted. In order to guarantee that the images were correctly classified, mislabeled images were also fixed. Additionally, because the images came from different directories on the Kaggle website, they were renamed for uniformity's sake. This procedure helps to increase the dataset's correctness.

3.3.2.2 Data Augmentation

The majority of CNN architectures demand uniformity in the size of all input data. For this reason, a collection of preprocessed images with precisely the same width and height makes up image recognition datasets. It becomes challenging to obtain a significant volume of different-sized brain tumor images for this study. Even with the collection of images from multiple common datasets, the size of the dataset did not indicate that high accuracy in model training would be possible. The total number of training images in the dataset was 8469, with 4304 and 4165 images in each of the two classes. However, brain scan includes a wealth of information, specifically, features that must be correctly retrieved. Additionally, because CNN is a data-driven methodology, the size of the dataset positively correlates with model accuracy. Data augmentation is a function that allows researchers to expand their datasets[29].

The process of producing new image variations using image transformations such as rotation, zooming, flipping, adjusting brightness, and contrast, and producing more image data from the preexisting dataset is known as data augmentation. The purpose of data augmentation is to expand and diversify the training dataset, which enhances the image classification model's strength and accuracy[5]. A variety of image augmentation methods were used to increase the dataset's robustness and diversity. Without changing the labels, the images were processed using the following methods.

- Rotation: Since little rotations between 0 and 20 degrees are particularly beneficial, the images were rotated left and right between 0 and 10 degrees around the axis.

- Translation: The images were rotated left, right, up, and down to prevent positional bias in the image data.
- Brightness Adjustment: The images' brightness was adjusted by adding and subtracting intensity values.
- Flipping: To create mirror images, the images were flipped both vertically and horizontally.
- Image cropping: The edges of the image that were superfluous and unimportant and could have an impact on the model's performance were cropped.

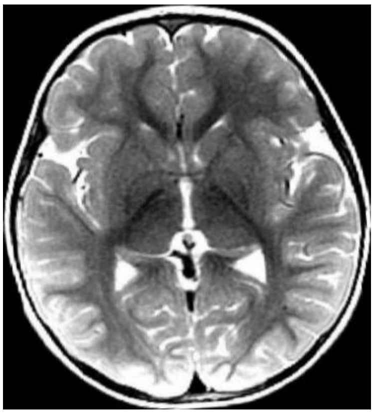


Figure 3.4: Original Image

Researcher's Concept, 2024

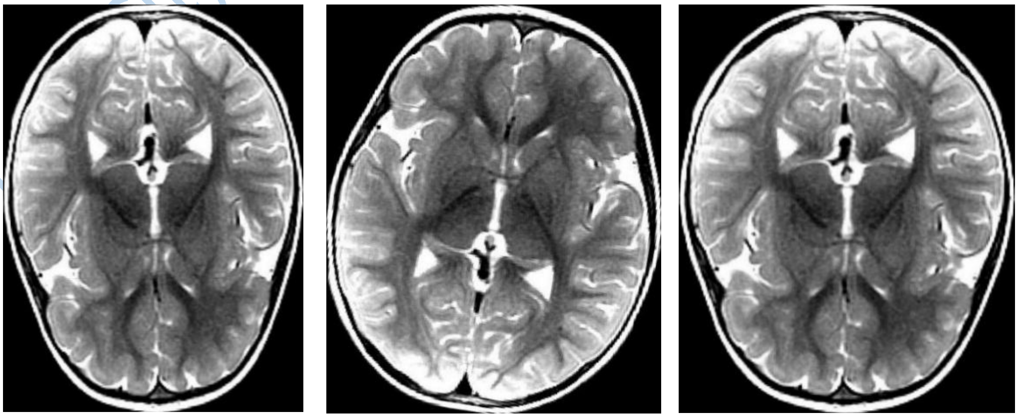


Figure 3.5: Augmented Images

Researcher's Concept, 2024

The total number of cancerous images increased to twenty-one thousand five hundred and seventeen (21517) and the number of non-cancerous images increased to twenty thousand eight hundred and twenty (20820) after applying data augmentation to the existing dataset. This resulted in a total of forty-two thousand, three hundred and thirty-seven (42337). 80% of the preprocessed images were selected for model training, 10% for system validation and the remaining 10% for testing the model.

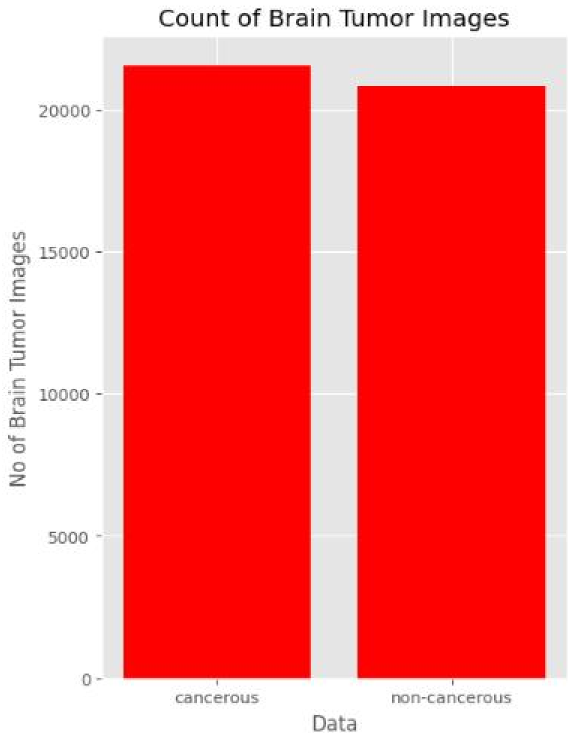


Figure 3.6: Histogram Showing the Number of Cancerous and Non-Cancerous Brain Tumor Images After Augmentation

Researcher's Concept, 2024

3.3.3 Building the Model

Convolutional neural networks (CNNs) are a type of deep neural network (DNN) that have shown outstanding performance in computer vision tasks, particularly in image classification. A unique kind of multi-layer neural network called a convolutional neural network (CNN or ConvNet) is modeled after the functioning of human visual and neural systems. Convolutional Neural Network (CNN) was created utilizing ideas from machine learning. Without assistance from humans, CNNs can autonomously learn and train from data. In actuality, using CNNs only requires a small amount of pre-processing. They create and modify their own image filters, which need to be properly written for the majority of models and algorithms. CNN frameworks consist of a number of layers and each layer carry out a certain task.

Input Layer: Each CNN starts with an input layer, where images or videos are captured, pre-processed, and then forwarded to the subsequent layers.

Convolution Layer: This layer uses learnable filters to take features out of images. This layer produces a feature map that shows which specific features are present in the input image and which ones are not.

Pooling Layer: This layer receives the extracted features from the convolution layer and then shrink the image while preserving the most crucial features. The most popular pooling operation is max pooling. Max pooling chooses the feature map's maximum value for each sub-region.

ReLU Layer: Rectified linear unit (ReLu) layers are activation functions used to reduce overfitting and improve the CNN's efficacy and accuracy. These layers make a model easier to train and yield more accurate results. ReLU makes all of the pooling layer's negative numbers equal to zero in order to preserve mathematical stability and prevent learned values from oscillating between 0 and infinity.

Fully Connected Layer: Using the output from the preceding layers, the fully connected layer creates the final classification. Every neuron in this layer is linked to every other neuron in layer[5].

In this study, AlexNet Convolutional Neural Network (CNN) model is modified to classify brain tumor MR image of varying sizes either as cancerous or non-cancerous. One of the first convolutional neural networks (CNNs) to be utilized for image identification and classification applications is AlexNet. It contains eight layers with parameters that can be learned. With the exception of the output layer, each of the model's five convolution layers uses a combination of max pooling, three fully connected layers, and ReLu activation function, which quickens the training process. Additionally, it included two dropout layers to stop its model from overfitting. Softmax is the activation function utilized in the output layer. The architecture has a total of 62.3 million parameters. Additionally, the ImageNet dataset is used to train the model. There are around 14 million images in the Imagenet dataset spread over 1,000 classes[31].

Because AlexNet has a deep architecture, the authors added padding to keep the feature map sizes from falling off too much. This model uses RGB images with dimensions of 227x227x3 as its input[31]. This suggests that all test images and all training set images must have a 227 x 227 pixel dimension. This also connotes all images has to be sized to 227x227 before utilizing them for training the network[32]. In 2012, AlexNet became victorious in the ImageNet large-scale visual recognition challenge (ILSVRC). The model was put forth by Alex Krizhevsky and his research team in their 2012 study titled, "ImageNet Classification with Deep Convolution Neural Network"[31].

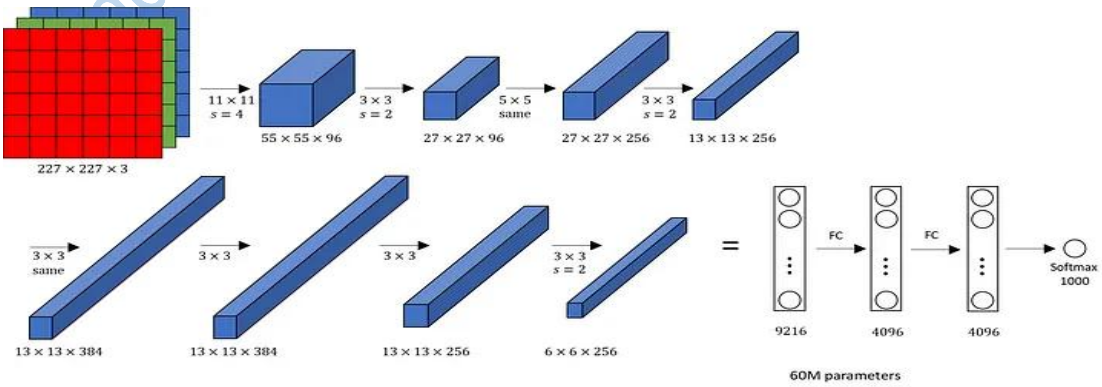


Figure 3.7: Architecture of the Alexnet Model[33]

Table 3.2: Summary of AlexNet Architecture

Layer	Filter/ Kernel	Kernel Size	Stride	Padding	Output Feature Map	Activation
Input image	-	-	-	-	227x227x3	-
Convolution 1	96	11 x 11	4	-	55 x 55 x 96	ReLU
Max Pooling 1	-	3 x 3	2	-	27 x 27 x 96	-
Convolution 2	256	5 x 5	1	2	27 x 27 x 256	ReLU
Max Pooling 2	-	3 x 3	2	-	13 x 13 x 256	-
Convolution 3	384	3 x 3	1	1	13 x 13 x 384	ReLU
Convolution 4	384	3 x 3	1	1	13 x 13 x 384	ReLU
Convolution 5	256	3 x 3	1	1	13 x 13 x 256	ReLU
Max Pooling 3	-	3 x 3	2	-	6 x 6 x 256	-
Dropout 1	Rate (0.5)	-	-	-	-	-
FC layer 1	-	-	-	-	9216	ReLU
Dropout 2	Rate (0.5)	-	-	-	-	-
FC layer 2	-	-	-	-	4096	ReLU
FC layer 3	-	-	-	-	4096	ReLU
Output	-	-	-	-	1000	Softmax

AlexNet Architecture[33]

The first convolution layer has 96 filters of size 11X11 with stride of 4. This layer uses ReLU as its activation function, and the resulting feature map has dimensions of 55x55x96. Additionally, in the output feature map, the number of filters becomes the channel. Next is the initial Maxpooling layer with a size of 3 by 3 and stride of 2. The resulting feature map has dimensions of 27x27x96. The second convolution layer has its filter reduced to 5x5 and there are 256 of such filters. One Stride and two padding were applied. Once more, ReLU serves as the activation function. The resulting output size is 27x27x256. Once more, a max-pooling layer of size 3x3 with stride of 2 was applied. The resulting feature map is of size 13x13x256. The third convolution operation has 384 filters of size 3x3, stride of 1 and padding of 1. Again, ReLU is employed as the activation function. The output feature map has dimension of 13x13x384. The fourth convolution operation has 384 filters of size 3x3. The stride and the padding is 1. Additionally, ReLU is employed as an activation function. The output size 13x13x384 remains constant.

The final convolution layer has 256 filters of size 3x3, the stride and padding are both set to one, and the activation function is ReLU. The resulting feature map is of size 13x13x256. The number of filters increases as the architecture goes deeper. This indicates that additional features are being extracted. Additionally, the filter size is dropping, indicating that the first filter was larger and that it will get smaller as more convolution layers are being applied, which will cause the feature map size to shrink. The third max-pooling layer which is of size 3x3 and stride 2 is applied resulting in the feature map with 6x6x256 dimension. After this is the first dropout layer. The default drop-out rate is set to 0.5. The first fully connected layer has a ReLU activation function and the output size is 9216. Another dropout layer with a predetermined dropout rate of 0.5 follows. A second fully connected layer with 4096 neurons with ReLU activation came after that. With 1000 neurons, the final fully connected layer, also known as the output layer, completes the data set with 1,000 classes. Softmax is the activation function that is employed at this layer. AlexNet architecture has a total of 62.3 million learnable parameters[31].

The key mathematical concepts involved in AlexNet architecture:

1. Convolutional Layers

Convolutional layers apply filters to the input image to create feature maps. The mathematical operation is a discrete convolution, which can be expressed as:

$$\text{Output}(i,j) = \sum_{m=0}^{k-1} \sum_{n=0}^{k-1} \text{Input}(i+m, j+n) \cdot \text{Filter}(m,n)$$

Where (k) is the filter size.

Here, (M) and (N) are the dimensions of the filter, and (i) and (j) are the spatial coordinates of the output.

2. Activation Function

AlexNet uses the ReLU (Rectified Linear Unit) activation function, which is defined as:

$$\mathrm{ReLU}(x) = \max(0, x)$$

This function introduces non-linearity into the model, allowing it to learn more complex patterns.

3. Pooling Layers

Pooling layers reduce the spatial dimensions of the feature maps. Max pooling, used in AlexNet, selects the maximum value from each patch of the feature map.

$$\mathrm{Output}(i, j) = \max_{m,n} \mathrm{Input}(i + m, j + n)$$

4. Normalization

Local Response Normalization (LRN) is used to normalize the output of the ReLU activation. The formula for LRN is:

$$b^i_{x,y} = a^i_{x,y} / \left(k + \alpha \sum_{j=\max(0,i-n/2)}^{\min(N-1,i+n/2)} (a^j_{x,y})^2 \right)^\beta$$

where i indicates the output of filter i, a(x,y), b(x,y) the pixel values at (x,y) position before and after normalization respectively, and N is the total number of channels. The constants (k,α,β,n) are hyper-parameters.

5. Fully Connected Layers

These layers perform a matrix multiplication followed by an addition of a bias term:

$$\text{Output} = \text{ReLU}(W \cdot \text{Input} + b)$$

where (W) is the weight matrix and (b) is the bias vector.

6. Softmax Function

The final layer uses the softmax function to convert the output into probabilities:

$$\text{Softmax}(z_i) = \frac{e^{z_i}}{\sum_j e^{z_j}}$$

This function ensures that the output values sum to 1, making them interpretable as probabilities.

Even though there are some more sophisticated and strong architecture (such as VGGNet, GoogLeNet/Inception, ResNet, and DenseNet) that outperform AlexNet on image classification tasks, they actually need a lot more computation, roughly ten times as much as AlexNet, which means they need more time and energy for training[35]. Nevertheless, AlexNet is still a treasured architecture that has achieved innovative performance on numerous image classification tasks.[34]. The selection of the CNN architecture is contingent upon the particular demands of the image classification problem, including the dataset's size and complexity, quantity of training data, and available processing resources for both training and inference[36].

3.3.3.1 The Modified AlexNet CNN Model

The modified AlexNet Convolutional Neural Network (CNN) model was built using Python and TensorFlow and it was implemented on a desktop computer with an Intel Core-i5 processor and 16 GB of RAM. The new model consists of six layers; five (5) convolutional layers and a global average pooling (GAP) layer that substitutes the fully connected layer which stops AlexNet from accepting images of varying sizes.

According to Table below, the procedure of the model began with the first convolution layer with 96 filters of size 11×11 to pull out the brain features. Then, to get the complete features from the feature maps MaxPooling with pool size 3×3 and batch normalization were applied. After this layer is the second convolution layer which is made up of 256 filters of size 5×5 and stride of 2. Like the first pooling layer, MaxPooling with pool size 3×3 and stride of 2 and batch normalization were applied. The third and fourth convolution layers comprises of 384 filters of size 3×3 each and padding of 1 and the fifth convolution layer contains 256 filters of size 3x3, padding of 1, MaxPooling with pool size 3×3, stride of 2. Batch normalization and dropout that deactivate 20% nodes arbitrarily were also inserted.

A layer called global average pooling was inserted after the last convolution layer to substitute the fully connected layer which doesn't accept variable input image size. In order to prevent partiality in the model towards the training data, two dense layers, each having 4096 nodes was formed. Preceding each dense layer is a dropout layer which randomly deactivates 50% of the active nodes. Rectified Linear Unit is the activation function used in the entire convolution layer as well as the dense layers. Lastly, the model has an output layer that makes use of sigmoid as the activation function and contains two nodes for each of the two classes of brain tumor. Binary crossentropy loss function and Stochastic Gradient Descent (SGD) optimization technique were used to compile the model.

Table 3.3: The Architecture of Modified AlexNet Model

Model Content	**Parameters**	**Input Size**	**Ouput Size**

First Convolution Layer	96 kernels of size 11x11, stride=4,	None, None, 3	None, None, 96
MaxPool	3x3, stride =2	None, None, 96	None, None, 96
Batch Normalisation		None, None, 96	None, None, 96
Second Convolution Layer	256 kernels of size 5x5, stride=2,	None, None, 96	None, None, 256
MaxPool	3x3, stride=2	None, None, 256	None, None, 256
Batch Normalisation		None, None, 96	None, None, 96
Third Convolution Layer	384 kernels of size 3x3 pad=1	None, None, 384	None, None, 384
Fourth Convolution Layer	384 kernels of size 3x3 pad=1	None, None, 384	None, None, 384
Fifth Convolution Layer	256 kernels of size 3x3 pad=1	None, None, 256	None, None, 256
MaxPool	3x3, stride =2	None, None, 96	None, None, 96
Batch Normalisation		None, None, 96	None, None, 96
Dropout	Deactivates 20% nodes randomly		
Global Average Pooling			
Dropout	Deactivates 50% nodes randomly		
First Dense Layer	4096 nodes, ReLU		
Dropout	Deactivates 50% nodes randomly		
Second Dense Layer	4096 nodes, ReLU		
Output	2 nodes for 2 classes, sigmoid		
Optimization Function	SGD and binary crossentropy loss function		

Researchers Concept, 2024

The mathematical concepts involved in modified AlexNet architecture for variable input size brain tumor images:

1. Convolutional Layers

Convolutional layers apply filters to the input image to create feature maps. The mathematical operation is a discrete convolution, which can be expressed as:

$$\text{Output}(i,j) = \sum_{m=0}^{k-1} \sum_{n=0}^{k-1} \text{Input}(i+m, j+n) \cdot \text{Filter}(m,n)$$

Where (k) is the filter size.

Here, (M) and (N) are the dimensions of the filter, and (i) and (j) are the spatial coordinates of the output.

2. Activation Function

AlexNet uses the ReLU (Rectified Linear Unit) activation function, which is defined as:

$$\text{ReLU}(x) = \max(0, x)$$

This function introduces non-linearity into the model, allowing it to learn more complex patterns.

3. Pooling Layers

Pooling layers reduce the spatial dimensions of the feature maps. Max pooling, used in AlexNet, selects the maximum value from each patch of the feature map.

$$\text{Output}(i,j) = \max_{m,n} \text{Input}(i+m, j+n)$$

4. Normalization

Local Response Normalization (LRN) is used to normalize the output of the ReLU activation. The formula for LRN is:

$$b^i_{x,y} = a^i_{x,y} / \left(k + \alpha \sum_{j=\max(0,i-n/2)}^{\min(N-1,i+n/2)} (a^j_{x,y})^2 \right)^\beta$$

where i indicates the output of filter i, a(x,y), b(x,y) the pixel values at (x,y) position before and after normalization respectively, and N is the total number of channels. The constants (k,α,β,n) are hyper-parameters.

5. Global Average Pooling

Global Average Pooling is a pooling operation designed to substitute fully connected layers in conventional Convolutional neural networks. It is a technique used in convolutional neural networks (CNNs) to lessen the spatial dimensions of feature maps. It works by taking the average value of each feature map, excellently producing a single value per feature map. This technique is particularly useful for handling variable image sizes in a consistent manner. Additionally, there is no parameter to optimize in the global average pooling thus the problem of overfitting is dodged at this layer.

Global Average Pooling can be expressed mathematically as:

$$GAPc = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} F_c(i,j)$$

where $F_c(i,j)$ is the value of the feature map at channel c and position (i,j).

6. Sigmoid Activation Function

The sigmoid function is defined mathematically as:

$$\frac{1}{(1+e^{\wedge}(-x))}$$

where x is the input value and e is the mathematical constant of 2.718. The function maps any input value to a value between 0 and 1, making it useful for binary classification and logistic regression problems. The range of the function is (0,1), and the domain is (-infinity,+infinity).

3.3.4 Training the Modified AlexNet CNN Model

Eighty percent of the images were chosen during the learning phase for system validation, ten percent for model testing, and the remaining ten percent for model training. SGD optimizer with 0.0001 learning rate is used to maximize the learning process. Given that overfitting is one of the variables in CNN training that cause worry, EarlyStopping is seen as a mechanism to halt CNN training when learning does not improve. The metric used to monitor the validation loss is called EarlyStopping. At the end of each training epoch, a model.fit() training loop checks to see if the validation loss is still decreasing after the patience of two intervals. The training ends when it is determined to be no longer decreasing, at which point model.stop_training is set to true. Additionally, ModelCheckpoint is applied to monitor the model's progress during and after training. After that, the checkpoints are preserved. This indicates that in the event of disruption, the model can pick up where it left off and save prolonged training time. When the validation accuracy stops increasing for more than patience of two interval, ReduceLROnPlateau is set to decrease the learning rate by 0.5.

Consequently, the learning rate is kept the same as long as it improves the validation accuracy but the learning rate is decreased when the results remain the same. All through the learning process, the model updates its weights and biases to reduce the given loss function by looping over the training data supplied as input for the specified number of epochs (10 epochs) with 500 steps per epoch. The essence of the validation data is to monitor the model's performance on hidden data.

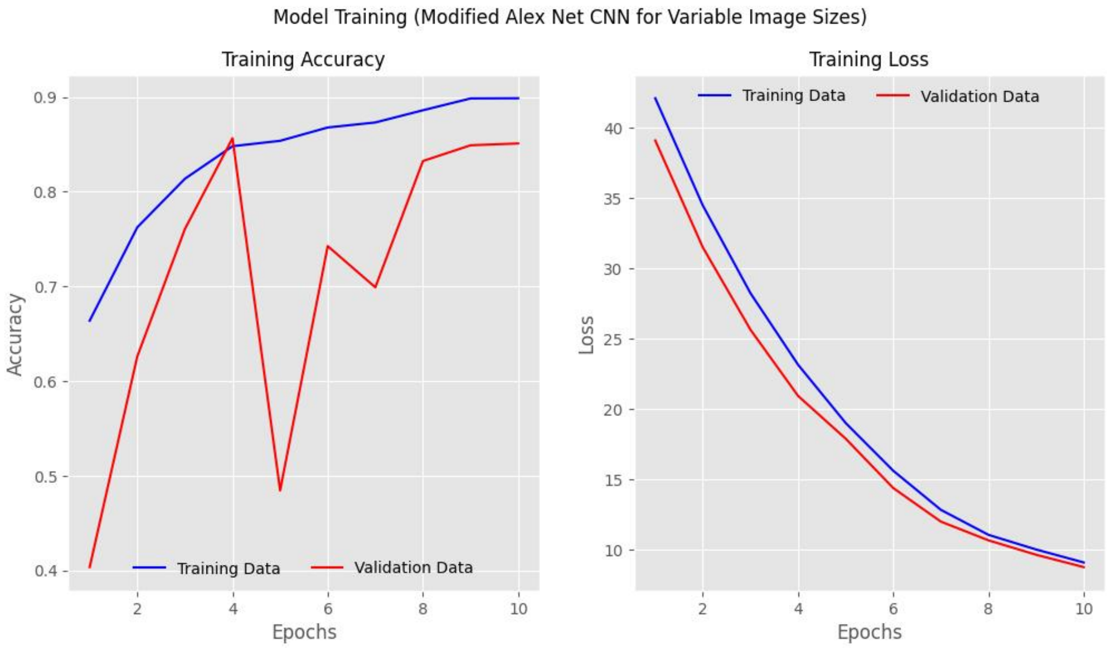


Figure 3.8: Accuracy and Loss Graphs of the Model

Researcher's Concept, 2024

The accuracy and loss graphs shown in the figure above demonstrate the model's performance. Y-axis shows the accuracy and loss, while the x-axis represents the number of model training epochs, or training cycles across the entire dataset. It can be seen from the graph that for some epochs, training accuracy is higher than validation accuracy. The training accuracy curve and the validation accuracy curve both trends upward as the number of epochs rises. Actually, the time span between one and four epochs is when there is dramatic increase. Validation accuracy drops between epochs 5 and 6, which led the ReduceLROnPlateau class to lower the learning rate to 0.005 in epoch 7 and 0.0025 in epoch 10, in order to enhance training performance. The model achieves 85.08% validation accuracy and 89.86% training accuracy at the end of training. There is not much of a difference between these two accuracies. This illustrates that the suggested model is not biased toward training images; rather, it classifies unknown images nearly as well.

3.3.5 Evaluation

The model was evaluated on test data and accuracy of 84.18% was achieved as shown in the figure below.

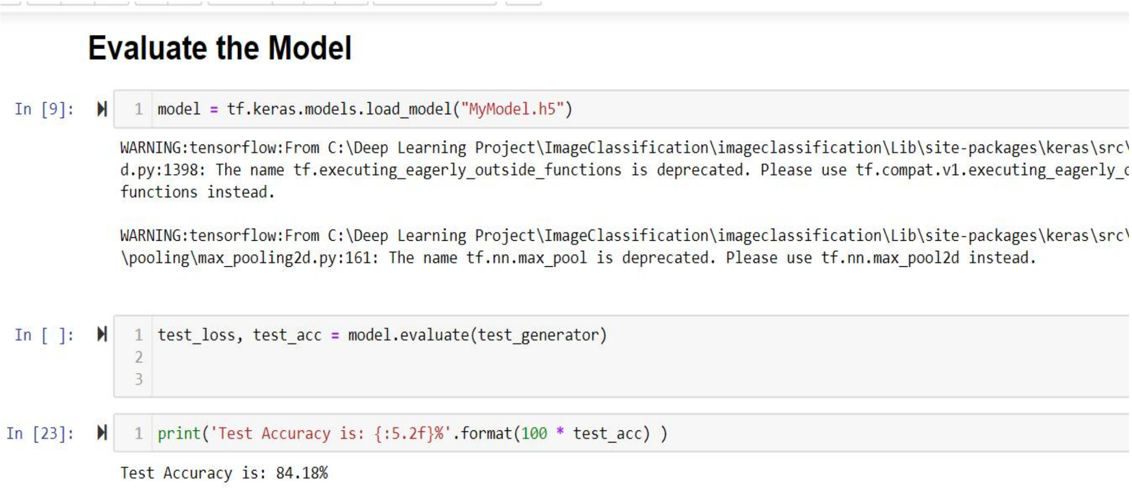


Figure 3.9: Model Evaluation

Researcher's Concept, 2024

3.3.6 Model Prediction

The prediction made by the model is shown in the figures below.

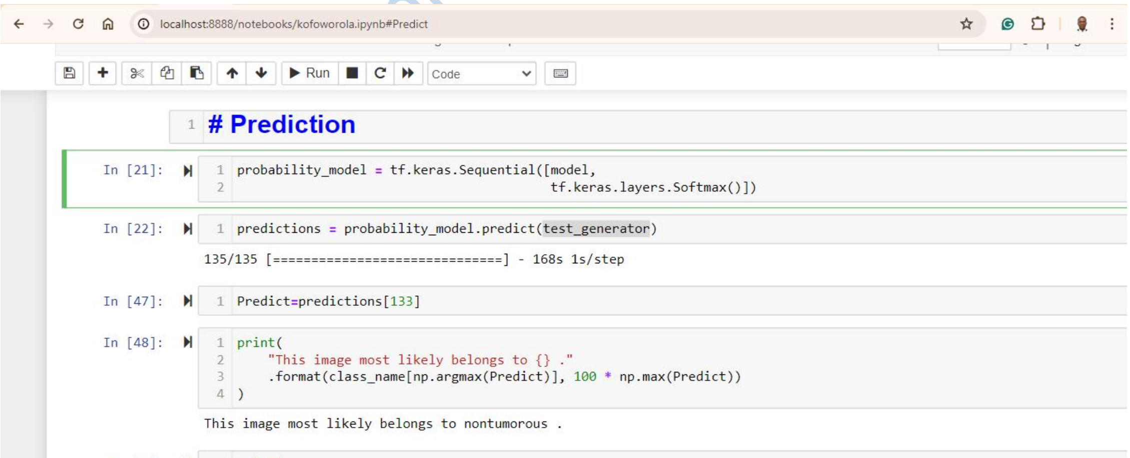


Figure 3.10: Prediction of the Non-Cancerous Brain Tumor Image

Researcher's Concept, 2024

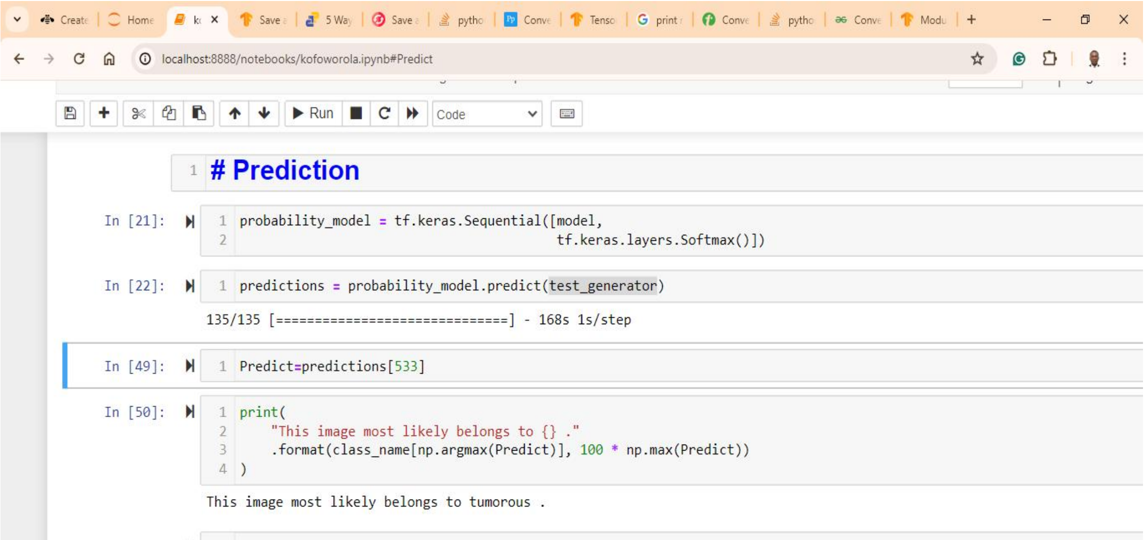


Figure 3.11: Prediction of the Cancerous Brain Tumor Image

Researcher's Concept, 2024

3.3.7 Model Performance Evaluation

In order to assess how well the modified AlexNet CNN model performed in classifying the brain tumor images, confusion matrix was generated. This matrix was then used to compare the predictions generated by the model with the original image labels. The number of MRI brain tumor images among the test images that are correctly classified is revealed by the confusion matrix. Because there are substantially more true negative and true positive values in the confusion matrix, it can be inferred that the classifier is appropriate for the dataset. In addition, a number of metrics, including the model accuracy, specificity, recall, precision, and F1 score, can be calculated using the values of the True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) on the confusion matrix as shown from equations 2.1 to 2.5.

The confusion matrix for the predicted output for testing and validation are displayed in Figures 3.12 and 3.13.

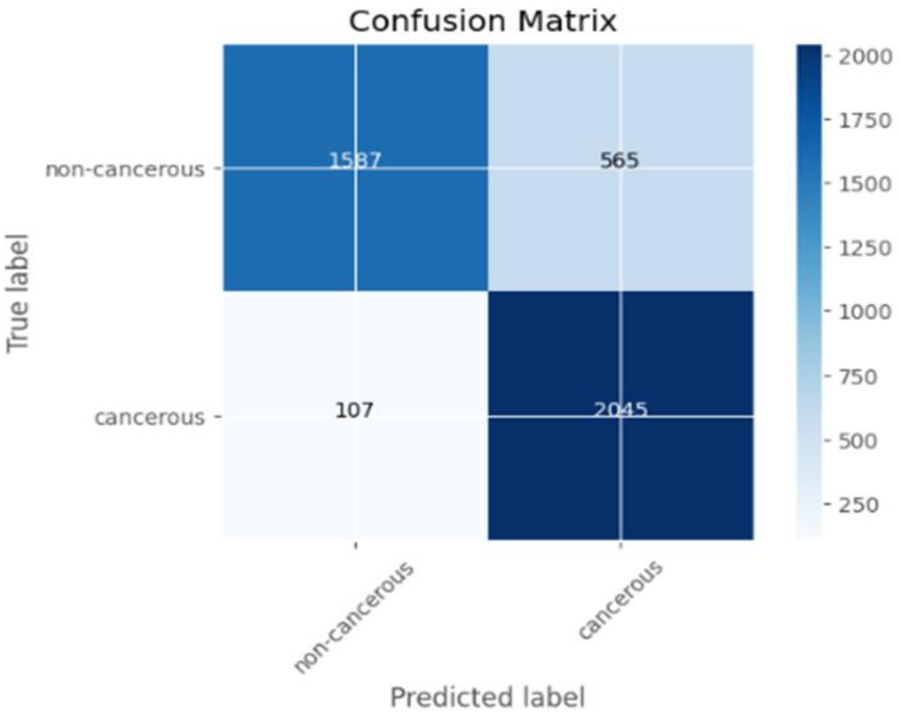


Figure 3.12: Confusion Matrix for the Test Dataset

Researcher's Concept, 2024

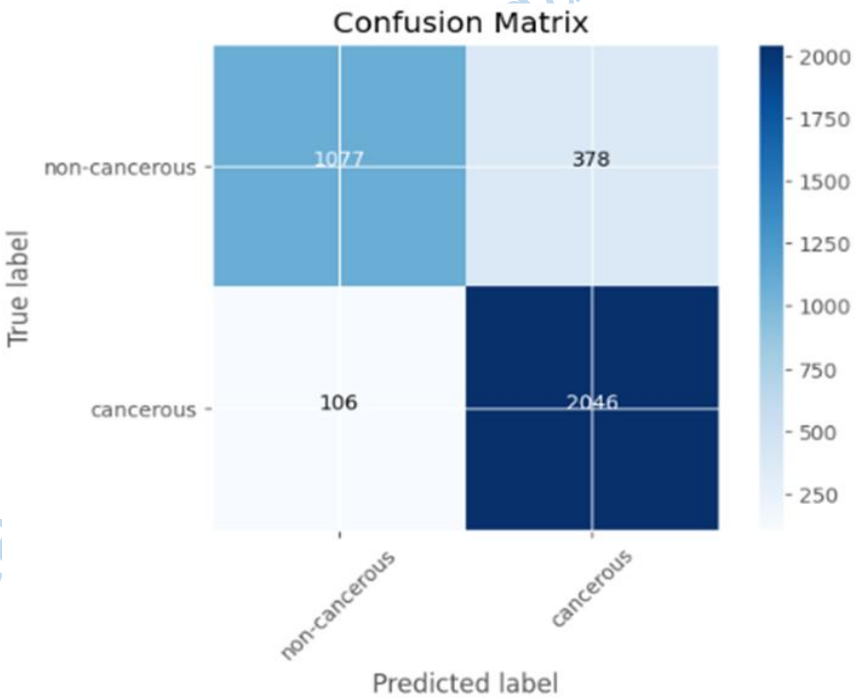


Figure 3.13: Confusion Matrix for the Validation Dataset

Researcher's Concept, 2024

The accuracy, precision, recall and f1-score of the model is given in Figure 3.14.

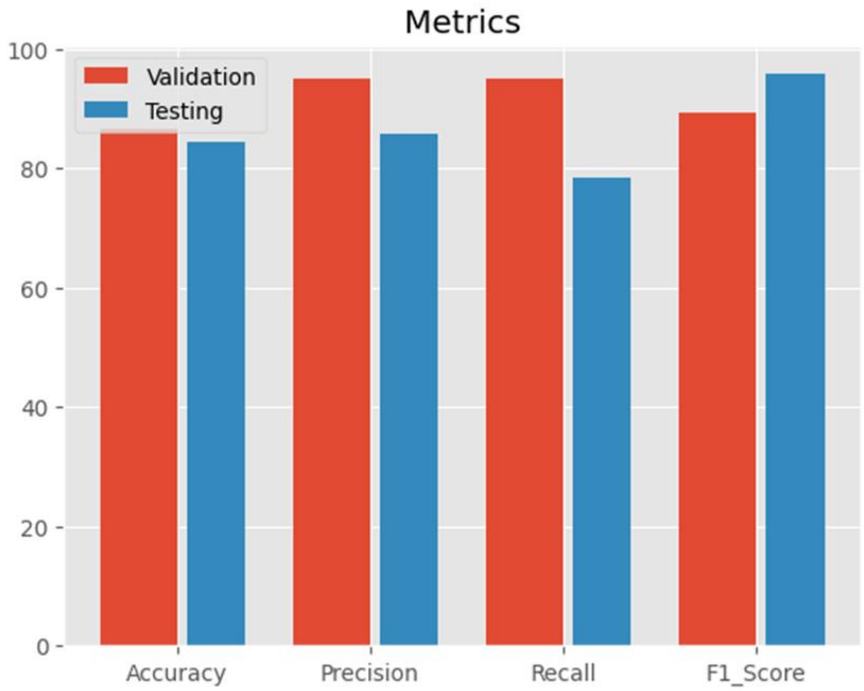


Figure 3.14: The Accuracy, Precision, Recall and F1-Score of the Model

Researcher's Concept, 2024

The modified AlexNet brain tumor CNN model is a powerful tool and its overall performance on test set achieves an accuracy of 84.39%, precision is78.35%, recall is 95.77%, and f1_score is 85.89%. Also, its overall performance on validation set achieves an accuracy of 86.58%, precision 84.41%, recall is 95.07%, and f1_score is 89.42%.

3.3.8 Model Framework

The overall architecture of the modified AlexNet model is shown in Figure 3.15 below. It is composed of data acquisition, data pre-processing, a CNN model, and performance measures.

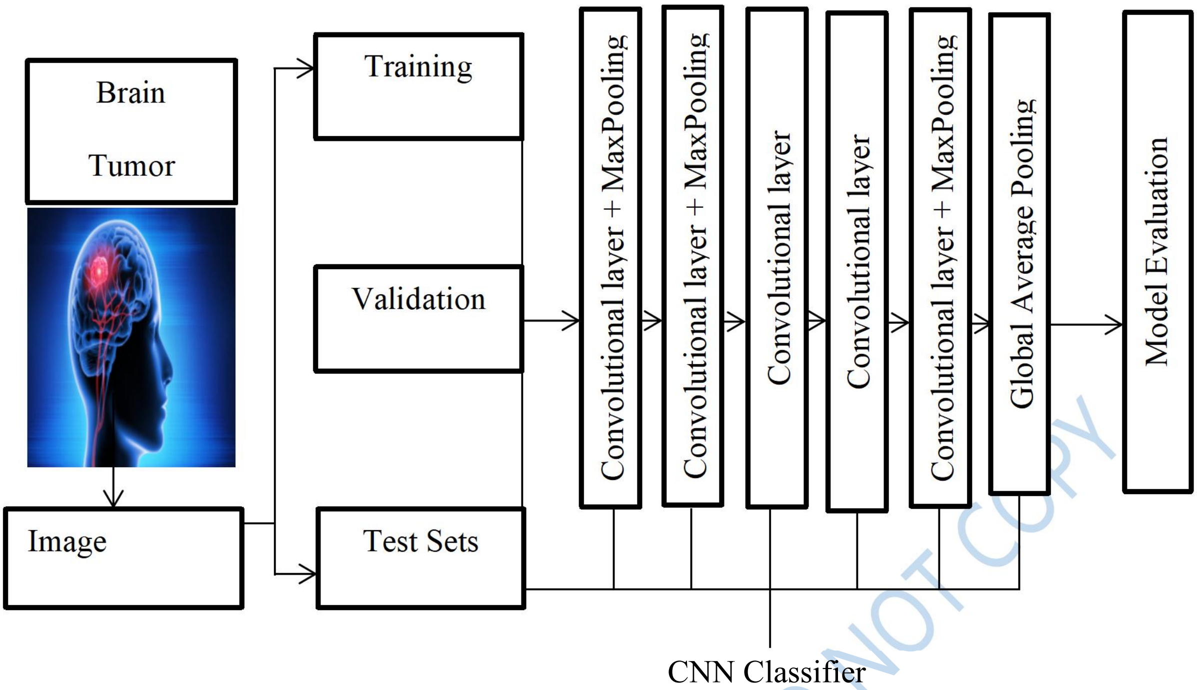


Figure 3.15: Block Diagram of the Model

Researcher's Concept, 2024

3.3.9 Algorithm for the Modified AlexNet Convolutional Neural Network (CNN) Classification Model

1. Import necessary libraries for the project
2. Unzip the zip file downloaded from Kaggle
3. Rename all the images for uniformity sake because the images were sourced from different directories on Kaggle site
4. Augment the data to increase the size of the image for the purpose of accuracy
5. Crop the unwanted black edge of the image
6. Split the image into the train set, validation set and test set images.
7. Set the class label for the mages (0 for non-cancerous tumor and 1 for cancerous tumor)
8. Normalize the Image
9. Build the modified AlexNet CNN Model
10. Compile the model.

11. Train the model

12. Plot the graph comparing the training accuracy and validation accuracy.

13. Evaluate the model using the test set.

14. Predict an image using the model

15. Draw the confusion matrix for actual output against the predicted output.

Endnotes

1. Y. O. Eun, & S. Donggil. *Developmental Research on An Interactive Application for Language Speaking Practice Using Speech Recognition Technology*. **Educational Technology Research and Development.** 69, 2021, 861–884.

2. A. R. Raymond. *Developmental Research for Information and technology*. **Studocu**. 2023.

3. S. Jarkko. *Technologies for Learning and Development*. 2023, https://sites.uef.fi/edtech/research-labs-and-focus-groups/ict-for-development/.

4. Quora. *Why is quantitative research important to an IT specialist?* 2023, https://www.quora.com/Why-is-quantitative-research-important-to-an-IT-specialist.

5. SuperAnnotate. *What Is Image Classification? Basics You Need to Know*. 2023, https://www.superannotate.com/blog/image-classification-basics.

6. B. Gaudenz. *A Complete Guide to Image Classification in 2023.* 2023, https://viso.ai/computer-vision/image-classification/.

7. S. Sorcha. *Image Classification in AI: How it works.* 2022, https://levity.ai/blog/image-classification-in-ai-how-it-works.

8. M. Silverio. *What's So Great About Jupyter Notebook?* 2023, https://builtin.com/software-engineering-perspectives/jupyter-notebook.

9. V. Singh. *Difference Between Jupyter Notebook and Python IDLE*. 2023, https://www.shiksha.com/online-courses/articles/jupyter-notebook-vs-python-idle/#:~:text=Jupyter%20Notebook%20is%20an%20interactive,writing%20and%20executing%20Python%20code.

10. Amazon Web Services. *What is an IDE (Integrated Development Environment)?* 2024, https://aws.amazon.com/what-is/ide/.

11. RedHat. *What is an IDE?* 2019, https://www.redhat.com/en/topics/middleware/what-is-ide.

12. 12 C. Steve. *Python Tutorial for Beginners: Learn Programming Basics [PDF] Guru99 2022.* 2022, https://www.guru99.com/.

13. K. Moez. *Convolutional Neural Networks: A Survey*. **Computers.** 12(8), 2023, 151.

14. geeksforgeeks. *Python Numpy*. 2023, https://www.geeksforgeeks.org/python-numpy/.

15. Numpy. *What is NumPy?* 2022, https://numpy.org/doc/stable/user/whatisnumpy.html#what-is-numpy.

16. J. B. Larson. *Python Image Processing: A Tutorial*. 2023, https://builtin.com/.

17. Neetika-khandelwal. *Image Processing in Python: Algorithms, Tools, and Methods You Should Know*. MLOps Blog. 2023, https://neptune.ai/blog/.

18. geeksforgeeks. *OpenCV Tutorial in Python*. 2024, https://www.geeksforgeeks.org/opencv-python-tutorial/?ref=shm.

19. G. Shreya. *Brain MRI Scans for brain tumor classification.* 2023, https://www.kaggle.com/datasets/shreyag1103/brain-mri-scans-for-brain-tumor-classification.

20. H. Ahmed. *Br35H :: Brain Tumor Detection 2020.* 2021, https://www.kaggle.com/datasets/ahmedhamada0/brain-tumor-detection.

21. A. P. G. Mikhael. *Br35H :: Brain Tumor Detection 2020.* 2022, https://www.kaggle.com/datasets/mikhaelapg/brain-tumor-detection.

22. N. Masoud. *Brain Tumor MRI Dataset.* 2021, https://www.kaggle.com/datasets/masoudnickparvar/brain-tumor-mri-dataset.

23. C. Navoneel. *Brain MRI Images for Brain Tumor Detection.* 2019, https://www.kaggle.com/datasets/navoneel/brain-mri-images-for-brain-tumor-detection?select=yes.

24. H. H. Mohammad. *Crystal Clean: Brain Tumors MRI Dataset.* 2023, https://www.kaggle.com/datasets/mohammadhossein77/brain-tumors-dataset.

25. G. Pratham. Brain_*Tumor_Classification*. 2023, https://www.kaggle.com/datasets/prathamgrover/brain-tumor-classification.

26. G. Shreya. *Brain MRI Scans for brain tumor classification.* 1311 brain tumor MRI scans belonging to four classes. 2023, https://www.kaggle.com/datasets/shreyag1103/brain-mri-scans-for-brain-tumor-classification.

27. D. Jean. *Brain MRI Images for Brain Tumor Detection*. 2020, https://www.kaggle.com/datasets/jjprotube/brain-mri-images-for-brain-tumor-detection.

28. Intelerad. *What is DICOM Image Format & Why is It Important in Radiology?* 2023, https://www.intelerad.com/en/2023/02/23/handling-dicom-medical-imaging-data/.

29. U. A. Tawsin, N.J. Mohammad, S.H. Mohammad, A. Karl, & S.H. Mohammed. *An Integrated Real-Time Deep Learning and Belief Rule Base Intelligent System to Assess Facial Expression under Uncertainty***.** Researchgate Conference Paper · June 2020, 1-7.

30. M. Eric**.** *Image Classification with Variable Input Resolution in Keras***. Mindboard. Mar 20, 2020.** Website, https://medium.com/mindboard/image-classification-with-variable-input-resolution-in-keras-cbfbe576126f; 2020.

31. S. Shipra. *Introduction to the Architecture of Alexnet.* 2023, https://www.analyticsvidhya.com/blog/2021/03/introduction-to-the-architecture-of-alexnet/.

32. P. Abhijeet. *Concept of AlexNet: - Convolutional Neural Network*. Analytics Vidhya. 2020, https://medium.com/analytics-vidhya/concept-of-alexnet-convolutional-neural-network-6e73b4f9ee30.

33. B. Siddhesh. *AlexNet Architecture Explained*. 2022, https://medium.com/@siddheshb008/alexnet-architecture-explained-b6240c528bd5.

34. geeksforgeeks. *Difference between AlexNet and GoogleNet*. 2023, https://www.geeksforgeeks.org/difference-between-alexnet-and-googlenet/.

35. Z. Sohaib. *Choosing the Right Pre-Trained Model: A Guide to VGGNet, ResNet, GoogleNet, AlexNet, and Inception*. 2023, https://medium.com/@sohaib.zafar522/choosing-the-right-pre-trained-model-a-guide-to-vggnet-resnet-googlenet-alexnet-and-inception-db7a8c918510#:~:text=AlexNet%20and%20ResNet%2D152%20both,more%20training%20time%20and%20energy.

36. C. Jonas. *7 Best Image Classification Models You Should Know in 2023*. 2023, https://jonascleveland.com/best-image-classification-models/#:~:text=Some%20of%20the%20most%20popular,resources%20for%20training%20and%20inference.

Chapter Four

Results and Discussion

4.1 Results

This chapter shows the result of the modified AlexNet Convolutional Neural Network (CNN) model from data preprocessing, model building, model training, model testing to model evaluation.

4.1.1 Pre-process the Acquired MRI Brain Tumor Images

4.1.1.1 Results of the Pre-processing

The process of applying techniques to enhance the quality of the image data and get it ready for the next phases is known as image preprocessing. The following were the main preprocessing activities applied on the brain tumor MR image obtained from Kaggle:

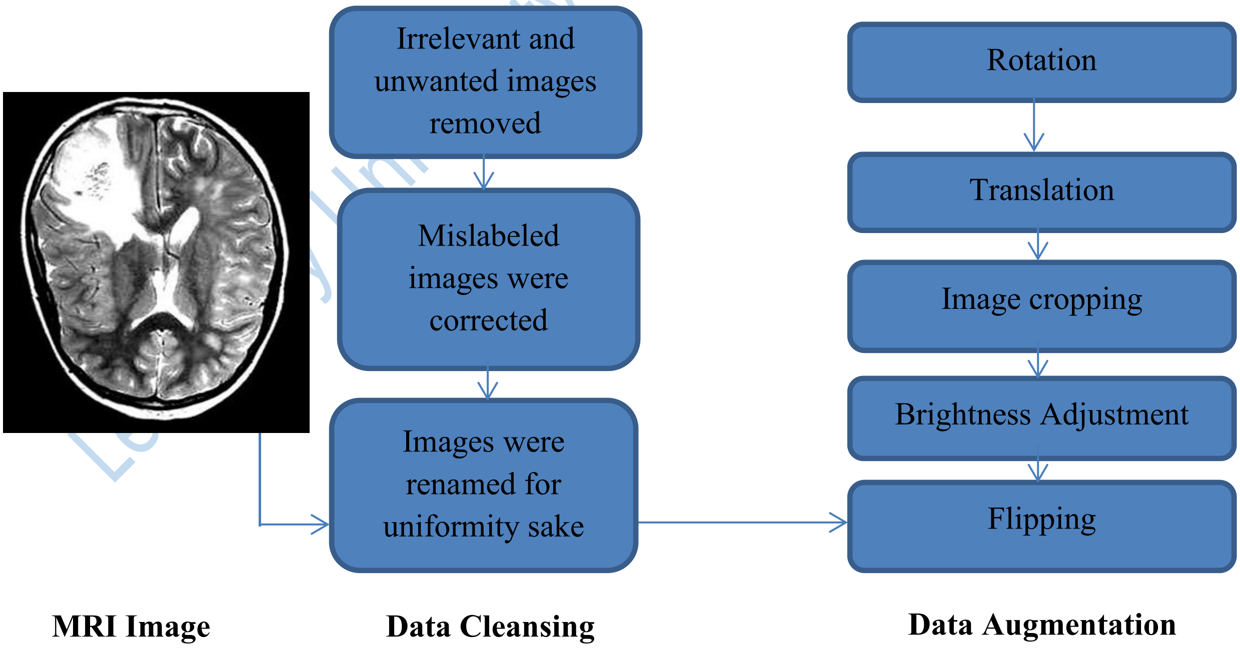


Figure 4.1: Flow Diagram of Image Preprocessing
Researcher's Concept, 2024

Data cleansing and data augmentation were the two preprocessing activities applied to the brain tumor MRI images. During the data cleansing, all unnecessary and undesired images were romoved and mislabeled images were corrected. Additionally, because the images came from different directories on the Kaggle website, they were renamed for uniformity's sake. This procedure is crucial because inaccurate data can result in inaccurate image classification models. The purpose of data augmentation is to expand and diversify the training dataset, which enhances the image classification model's strength and accuracy. image augmentation methods that were used to increase the dataset's robustness and diversity are rotation, translation, brightness adjustment, flipping and image cropping.

Figure 4.2: Original Image

Researcher’s Concept, 2024

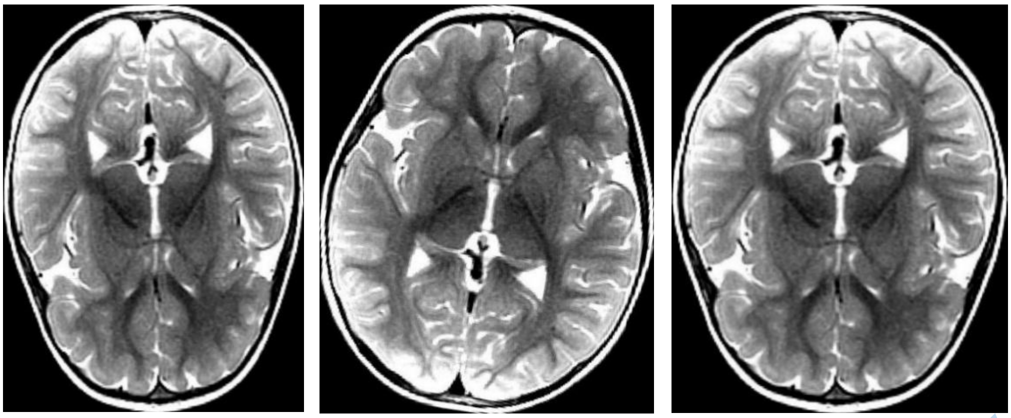


Figure 4.3: Augmented Images

Researcher's Concept, 2024

4.1.2 Develop the CNN Model and Classify the Brain Tumor

The new model consists of six layers; five (5) convolutional layers and a global average pooling (GAP) layer that substitutes the fully connected layer which stops AlexNet from accepting images of varying sizes. In order to prevent partiality in the model towards the training data, two dense layers, each having 4096 nodes was formed. Preceding each dense layer is a dropout layer which randomly deactivates 50% of the active nodes. Rectified Linear Unit is the activation function used in the entire convolution layer as well as the dense layers. Lastly, the model has an output layer that makes use of sigmoid as the activation function and contains two nodes for each of the two classes of brain tumor. Binary crossentropy loss function and Stochastic Gradient Descent (SGD) optimization technique were used to compile the model.

4.1.2.1 Python Code for Building the Modified AlexNet Model

```
import tensorflow as tf
from tensorflow.keras import layers, models
from keras import regularizers
#def global_average_pooling(x):
def reduce_mean(x):
   return tf.reduce_mean(x, axis=[1, 2])
def Modified_Alexnet(input_shape, num_classes):
```

```
inputs = layers.Input(shape=input_shape)
# Layer 1
x = layers.Conv2D(96, 11, strides=4, padding='valid', activation='relu')(inputs)
x = layers.MaxPooling2D(3, strides=2)(x)
x = layers.BatchNormalization()(x)
# Layer 2
x = layers.Conv2D(256, 5, padding='same', activation='relu')(x)
x = layers.MaxPooling2D(3, strides=2)(x)
x = layers.BatchNormalization()(x)
# Layer 3
x = layers.Conv2D(384, 3, padding='same', activation='relu')(x)
# Layer 4
x = layers.Conv2D(384, 3, padding='same', activation='relu')(x)
# Layer 5
x = layers.Conv2D(256, 3, padding='same', activation='relu')(x)
x = layers.MaxPooling2D(3, strides=2)(x)
x = layers.BatchNormalization()(x)
x = layers.Dropout(0.2)(x)
# Global Average Pooling
#x = layers.GlobalAveragePooling2D()(x)
x = reduce_mean(x)
# Fully Connected Layers (removed)
# x = layers.Flatten()(x)
# x = layers.Dense(4096, activation='relu')(x)
# x = layers.Dropout(0.5)(x)
# x = layers.Dense(4096, activation='relu')(x)
# Regularization
x = layers.Dropout(0.5)(x)
x = layers.Dense(4096, activation='relu', kernel_regularizer=regularizers.l2(0.01))(x)
x = layers.Dropout(0.5)(x)
x = layers.Dense(4096, activation='relu', kernel_regularizer=regularizers.l2(0.01))(x)
# Output Layer
outputs = layers.Dense(num_classes, activation='sigmoid')(x)
model = models.Model(inputs, outputs)
```

```
    return model
input_shape = (None, None, 3) # Variable image size
num_classes = 2 # Number of classes
model= Modified_Alexnet (input_shape, num_classes)
```

4.1.2.2 The Modified AlexNet CNN Model Training Process

During the learning process, 80% of the preprocessed images were chosen as model training, 10% for model validation, and the remaining 10% for model testing. SGD optimizer with 0.0001 learning rate were used to maximize the learning process. EarlyStopping, Model.fit() training loop, Model.stop_training and ReduceLROnPlateau were some of the regularizations that were used to prevent the model from overfitting. All through the learning process, the model updates its weights and biases to reduce the given loss function by looping over the training data supplied as input for the specified number of epochs (10 epochs) with 500 steps per epoch. The essence of the validation data is to monitor the model's performance on hidden data. The snippet of the training process is shown in the figure below.

```
filepath = 'model12042024.h5'
es = EarlyStopping(monitor='val_loss', verbose = 1, mode='min',patience=2)
cp = ModelCheckpoint(filepath, monitor='val_loss', verbose = 1, save_best_only=True, save_wei
ghts_only=False, mode='auto',save_freq='epoch')
lrr = ReduceLROnPlateau(monitor='val_accuracy', patience=3, verbose = 1, factor = 0.5, min_lr
= 0.0001)
history = model.fit(train_generator,steps_per_epoch=500, epochs = 10, validation_data=valid_ge
nerator, callbacks=[es,cp,lrr])
Epoch 1/10
500/500 [============================] - ETA: 0s - loss: 42.0853 - accuracy: 0.6637
Epoch 1: val_loss improved from inf to 39.08426, saving model to model12042024.h5
500/500 [==============================] - 1684s 3s/step - loss: 42.0853 - accuracy:
0.6637 - val_loss: 39.0843 - val_accuracy: 0.4034 - lr: 0.0100
Epoch 2/10
500/500 [============================] - ETA: 0s - loss: 34.4546 - accuracy: 0.7623
Epoch 2: val_loss improved from 39.08426 to 31.47517, saving model to model12042024.h5
500/500 [==============================] - 1831s 4s/step - loss: 34.4546 - accuracy:
0.7623 - val_loss: 31.4752 - val_accuracy: 0.6260 - lr: 0.0100
Epoch 3/10
500/500 [============================] - ETA: 0s - loss: 28.2204 - accuracy: 0.8136
Epoch 3: val_loss improved from 31.47517 to 25.62425, saving model to model12042024.h5
500/500 [==============================] - 1903s 4s/step - loss: 28.2204 - accuracy:
0.8136 - val_loss: 25.6242 - val_accuracy: 0.7605 - lr: 0.0100
Epoch 4/10
```

Figure 4.4: The Modified AlexNet CNN Model Training Process

Researcher's Concept, 2024

4.1.2.3 Accuracy and Loss Graph of the Model

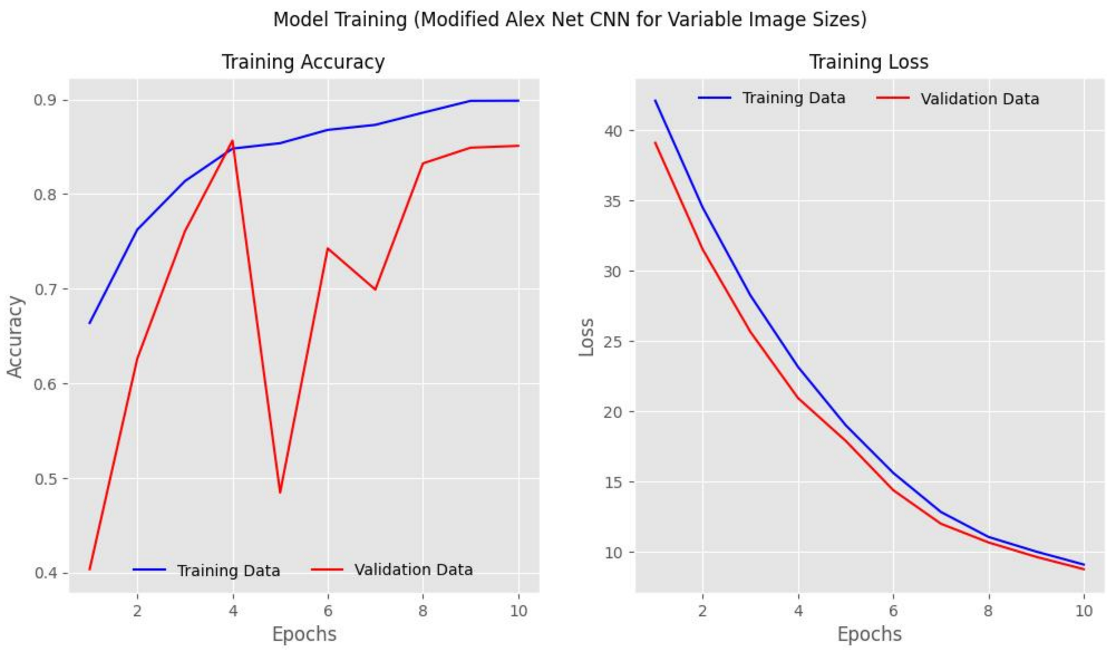


Figure 4.5: Accuracy and Loss Graphs of the Model

Researcher's Concept, 2024

The accuracy and loss graphs shown in the figure above demonstrate the model's performance. X-axis represents the number of model training epochs while Y-axis shows the accuracy and loss, The model achieved 85.08% validation accuracy and 89.86% training accuracy at the end of training.

4.1.2.4 Model Testing

The model was evaluated on the test data and accuracy of 84.18% was achieved as shown in the Python Code below.

```
model = tf.keras.models.load_model("model10EpochsWithNewDataSet.h5")
test_loss, test_acc = model.evaluate(test_generator)
135/135 [==============] - 108s 803ms/step - loss: 2.1992 - accuracy: 0.8436
print('Test Accuracy is: {:5.2f}%'.format(100 * test_acc) )
Test Accuracy is: 84.18%
```

4.1.2.5 Model Prediction

The model's prediction is displayed in the figures below. Two types of results were generated: the first one is the non-cancerous classification and the second one is cancerous portion after classification.

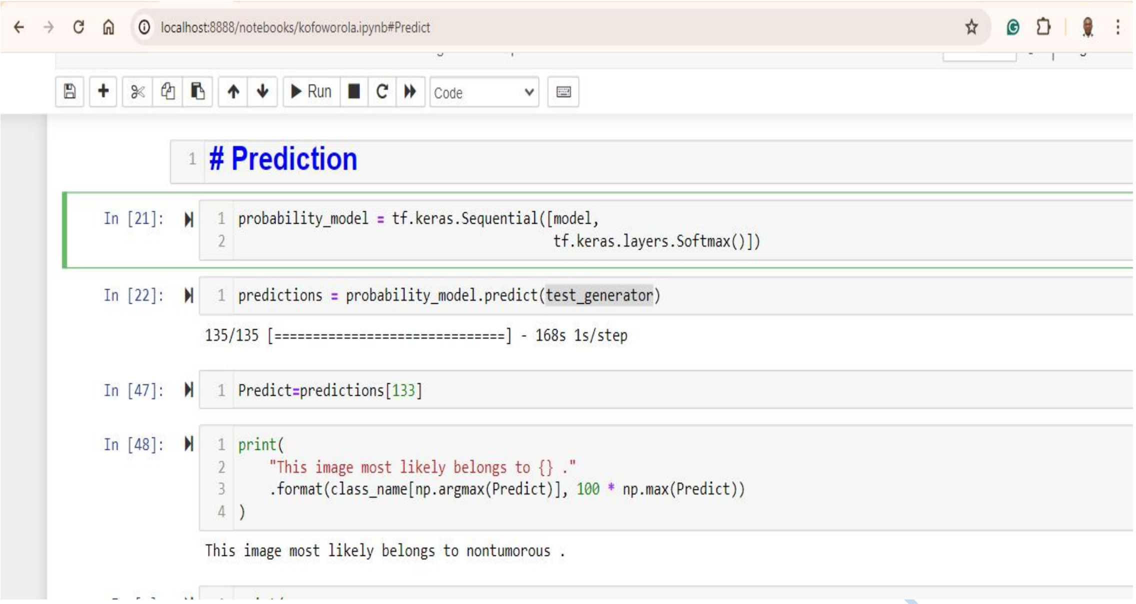


Figure 4.6: Prediction of the Non-Cancerous Brain Tumor Image

Researcher's Concept, 2024

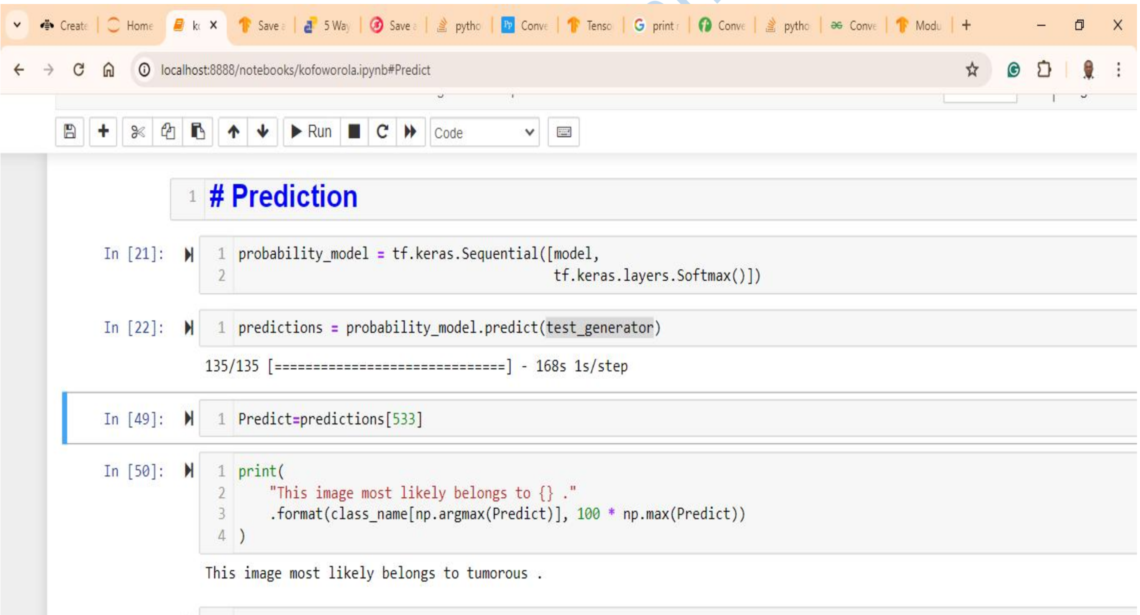


Figure 4.7: Prediction of the Cancerous Brain Tumor Image

Researcher's Concept, 2024

4.1.3 Evaluation of the Model's Performance

A confusion matrix is a tool for evaluating the performance of a classification model by comparing its predicted labels to the truth label. It displays the number of True positive (TP), True negative (TN), False positive (FP) and False negative (FN) of the model's prediction. Using the TP, TN, FP and FN, we can calculate various classification metrics such as accuracy, precision, recall and f1 score of a model. The confusion matrices for the predicted output for model testing and validation are displayed in figures 4.8 and 4.9 respectively.

For the test data:

TP = 1587: This indicates that 1587 of the brain tumor images were correctly predicted as cancerous by the model.

TN = 2045: This indicates that 2045 of the brain tumor images were correctly predicted as non-cancerous by the model.

FP = 565: This indicates that 565 of the brain tumor images were incorrectly predicted as cancerous by the model.

FN = 107: This indicates that 107 of the brain tumor images were incorrectly predicted as non-cancerous by the model.

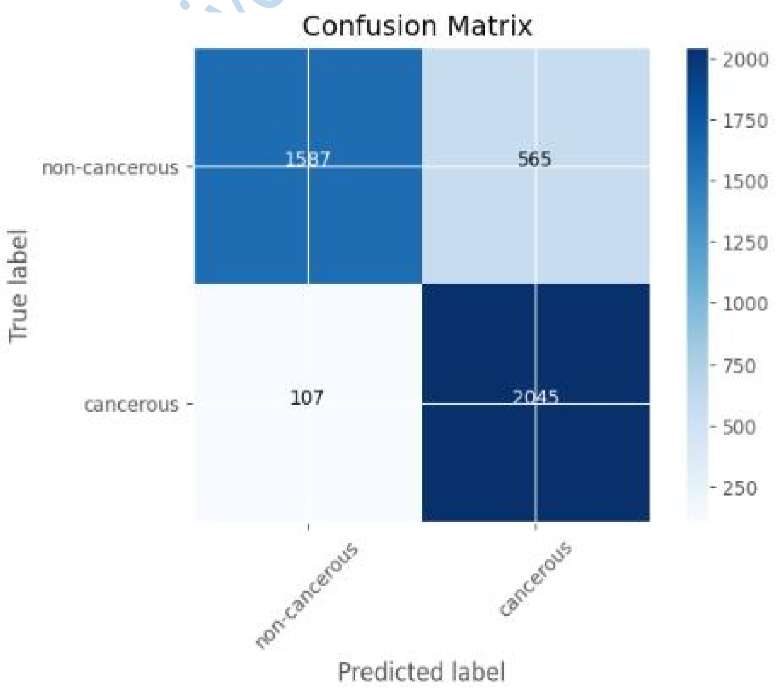


Figure 4.8: Confusion Matrix for the Test Dataset

Researcher's Concept, 2024

For the validation data:

TP = 1077: This indicates that 1077 of the brain tumor images were correctly predicted as cancerous by the model.

TN = 2046: This indicates that 2046 of the brain tumor images were correctly predicted as non-cancerous by the model.

FP = 378: This indicates that 378 of the brain tumor images were incorrectly predicted as cancerous by the model.

FN = 106: This indicates that 106 of the brain tumor images were incorrectly predicted as non-cancerous by the model.

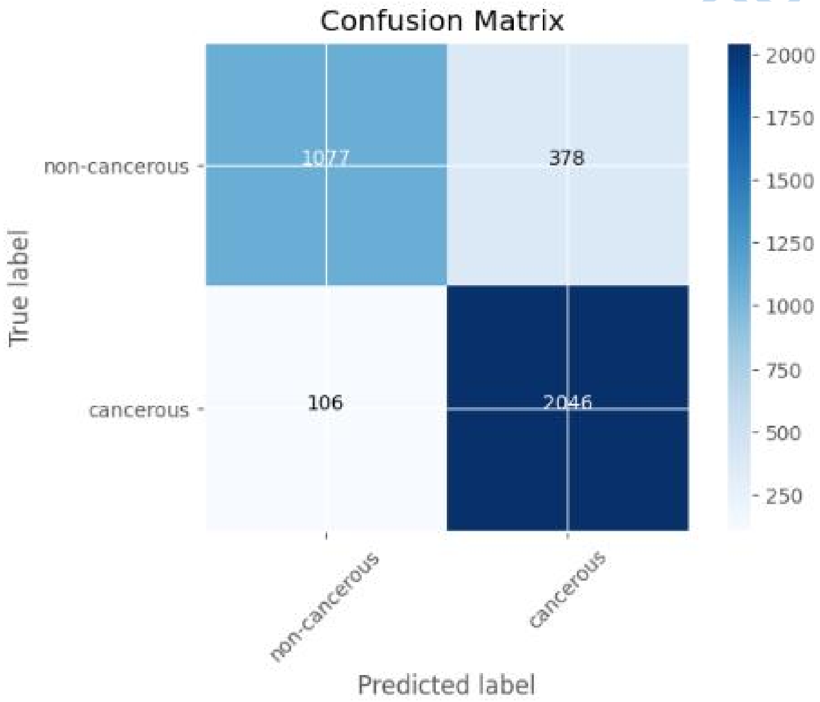


Figure 4.9: Confusion Matrix for the Validation Dataset
Researcher's Concept, 2024.

From the confusion matrixes, the accuracy, precision, recall and f1-score of the model were evaluated as shown in figure 4.10 below. For the test data, an accuracy which calculate the proportion of the correctly classified instances is 84.39%. The precision which is 78.35% shows the number of correctly predicted positive instances out of all instances that are predicted as positive. The recall which shows the number of positive instances successfully identified by the model is 95.77% and lastly, the f1 score which calculates the harmonic average between recall and precision rates is 85.89%.

For the validation data, an accuracy which indicates the ratio of correctly predicted instances to the total number of samples evaluated is 86.58%. The precision which is 84.41%, shows the number of correctly predicted positive instances out of all instances that are predicted as positive. The recall which shows the number of positive instances successfully identified by the model is 95.07% and lastly, the f1 score which calculates the harmonic average between recall and precision rates is 89.42%.

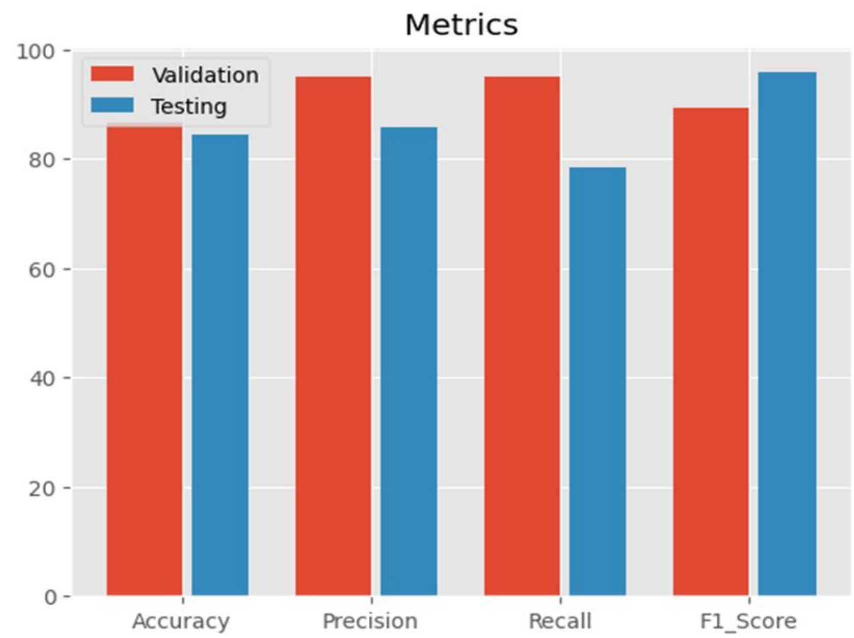


Figure 4.10: The Accuracy, Precision, Recall and F1-Score of the Model

Researcher's Concept, 2024

4.2 Discussion of Results

For the purpose of detecting and classifying brain tumors, several convolutional neural network models, including AlexNet VGGNet, GoogleNet, ResNet, DenseNet, and many more, have been developed. Furthermore, a large number of researchers had retrained pre-existing CNN models to speed up computation time and enhance performance. The issue with nearly all of these current networks is that their input sizes are fixed. The majority of researchers that have used these CNN models for brain tumor detection and classification closely follow the model specifications. The brain tumor images are cropped or warped to fit them into CNN models. The images are then classified, and the model's performance is assessed and they stated in their conclusion that their models outperform current models but to the best of the researcher's knowledge, no research has been done on the use of CNN for brain tumor detection and classification that has taken into account finding a solution to the fixed input size issue that is related to CNN models. Therefore, the goal of this work is to modify AlexNet convolutional neural network architecture to accept different sizes of brain tumor MR images as input and then classify the images as cancerous or non-cancerous.

In this study we developed a brain tumor-oriented CNN model that accepts varying input size. The model was trained with brain tumor MRI images of varying sizes collected from various repositories on Kaggle. To the best of the researchers' knowledge, no such CNN model has been developed for only an application area or an area in medical field. Presently, the researchers can only guarantee the effectiveness of this model on brain tumor MRI images. Its efficacy on other image recognition dataset like animals, human faces and the likes may be communicated in future research.

In order to meet the study's objectives, the brain tumor images that were gathered from Kaggle were preprocessed to enhance the quality of the image data. The two main pre-processing techniques applied on the brain tumor images were data cleansing and data augmentation. After the application of data augmentation techniques to the preexisting dataset, the total number of cancerous images rises to 21,517 and non-cancerous images rise to 20,820 making a total of 42,337 brain tumor images. The modified AlexNet

Convolutional Neural Network (CNN) model was built using Python and TensorFlow and it was implemented on a desktop computer with an Intel Core-i5 processor and 16 GB of RAM. The new model consists of six layers; five (5) convolutional layers and a global average pooling (GAP) layer that substitutes the fully connected layer which stops AlexNet from accepting images of varying sizes. The binary crossentropy loss function and stochastic gradient descent optimization approach are used to compile the model.

Eighty percent of the preprocessed brain tumor images were chosen during the learning phase for system validation, ten percent for model testing, and the remaining ten percent for model training. SGD optimizer with 0.0001 learning rate was used to maximize the learning process. Given that overfitting is one of the variables in CNN training that cause worry, EarlyStopping is seen as a mechanism to halt CNN training when learning does not improve. The metric used to monitor the validation loss is called EarlyStopping. At the end of each training epoch, a model.fit() training loop checks to see if the validation loss is still decreasing after the patience of two intervals. The training ends when it is determined to be no longer decreasing, at which point model.stop_training is set to true. Additionally, ModelCheckpoint is applied to monitor the model's development during and after training. After that, the checkpoints are preserved. This indicates that in the event of disruption, the model can pick up where it left off and save prolonged training time. When the validation accuracy stops increasing for more than patience of two interval, ReduceLROnPlateau is set to decrease the learning rate by 0.5. Consequently, the learning rate is kept the same as long as it improves the validation accuracy but the learning rate is decreased when the results remain the same. All through the learning process, the model updates its weights and biases to reduce the given loss function by looping over the training data supplied as input for the specified number of epochs (10 epochs) with 500 steps per epoch. The essence of the validation data is to monitor the model's performance on hidden data. The model achieves 85.08% validation accuracy and 89.86% training accuracy at the end of training. An accuracy of 84.18% was attained when the model was assessed using test data.

The performance of the model was evaluated using confusion matrix which gives details on the number of MRI brain tumor images that are properly classified among the test images. From the confusion matrix, it was deduced that the classifier is suitable for the dataset by considering the comparatively greater number of true negative and true positive values. The modified AlexNet brain tumor CNN model is a powerful tool and its overall performance on test set achieves an accuracy of 84.39%, precision is78.35%, recall is 95.77%, and f1_score is 85.89%. Also, its overall performance on validation set achieves an accuracy of 86.58%, precision 84.41%, recall is 95.07%, and f1_score is 89.42%.

Chapter Five

Conclusion

5.1 Summary of Results

CNN is a powerful tool in the field of disease detection and classification that has been utilized for brain tumor identification and classification. The potential of CNN in medical field has drawn the attention of many researchers. Patients can receive therapy more quickly and with greater precision when CNNs are used to predict brain tumors from MRI scans. The radiologist and doctors can make swift decisions based on this prediction. While CNNs have shown remarkable achievements across a wide range of tasks and domains, their sensitivity to input size is still a significant issue that restricts their practical applications.

In order to solve this problem, three specific objectives were designed. Firstly, to pre-process the acquired MRI brain tumor images. Secondly, to implement a CNN model that will accept varying sizes of MRI brain tumor images and then classify the brain tumor as cancerous and non-cancerous and lastly, to evaluate the performance of the model in terms of accuracy, precision, recall and f1 score.

The first obejective was achieved by applying data cleansing and data augmentation techniques to the acquired brain tumor MRI images. The essence of data cleansing is to increase the dataset's correctness while data augmentation was used to expand and diversify the training dataset to enhance the image classification model's robustness and accuracy.

To achieve the second objective, AlexNet CNN architecture was modified to accept different sizes of brain tumor images and classify them as either cancerous or non-cancerous. This modified AlexNet model consist of six layers; five convolutional layer and a Global Average Pooling layers that replaces the fully connected layer which stops AlexNet from accepting images of varying sizes. The datasets for this study consist of 42,337 preprocessed brain tumor images out of which 21517 are cancerous and remaining 20820 are non-cancerous. Ten percent of the preprocessed images were used for model validation, ten percent were used for model testing, and eighty percent were used for model training. The model achieves 85.08% validation accuracy and 89.86% training accuracy at the end of training. After evaluating the model with test data, an accuracy of 84.18% was attained.

To achieve the third objective, confusion matrices were generated to assess the performance of the model. By taking into account the relatively higher number of true negative and true positive values, it was inferred from the confusion matrix that the classifier is appropriate for the dataset. From the confusion matrixes, the accuracy, precision, recall and f1-score of the model were evaluated. The programming tools used for building this model were Python and TensorFlow and it was implemented on a desktop computer with Intel Core-i5 processor and 16 GB RAM.

5.2 Recommendations

The development of CNN models that accept various sizes of brain tumor images is a critical advancement in medical image analysis. Not only does it promise to improve diagnostic performance and model robustness, but it also aligns with ongoing efforts to streamline and optimize healthcare technologies. The potential benefits for both clinicians and patients are immense, making this research highly relevant and impactful. This work could lead to multiple avenues for future research such as investigating the integration of data from different imaging modalities (e.g., MRI, CT, PET scans) that have varying image sizes and resolutions. Applying the model in a clinical setting where real-time tumor detection is required, especially in intraoperative or emergency situations and using data augmentation strategies that focus on simulating real-world variations in image sizes to further enhance model robustness.

5.3 Contribution to Knowledge

One of the most significant advancements in the application of Convolutional Neural Networks (CNNs) for brain tumor detection and classification is the development of models that can accept input images of varying sizes. This capability contributes to the field of medical imaging by enhancing the robustness, adaptability, and efficiency of CNN-based systems in clinical practice. These models provide solutions to several challenges in the field, such as image variability, data preprocessing, multi-resolution data handling, and multi-modal integration. By allowing for the direct processing of images of different sizes, these CNN architectures are able to improve diagnostic accuracy, reduce preprocessing burdens, and offer more scalable and flexible solutions for real-world clinical environments. This, in turn, leads to more accurate, faster, and reliable tumor detection, with potential to significantly improve patient outcomes in brain tumor diagnosis.

The ability to accept images of various sizes broadens the applicability of CNN models in real-world clinical scenarios, where data variability is common. This flexibility allows for greater scalability in deploying AI-driven diagnostic tools, ensuring that models can be used universally across hospitals, imaging centers, and research studies. This also means that the model can be trained on larger, more diverse datasets, leading to a more generalizable and reliable system for brain tumor detection.

5.4 Suggestion for Further Studies

At the end of building and training this modified AlexNet model, the following suggestions for further works were made:

i. This study may be reproduced and deep-trained for improved clinical experience by increasing the number of training epochs to increase the model's accuracy.

ii. Supplying a massive high-resolution brain tumor image dataset can lead to a significantly higher accuracy and reduce overfitting.

iii. The accuracy of the model can further be increased with classifier boosting approaches, making this tool a valuable asset for any medical facility treating brain tumors.

iv. The model can be reproduced using powerful CNN architecture such as GoogLeNet.

v. The need for the creation of CNN models that can process brain MRI images of different sizes, classify the brain MRI images as cancerous and non-cancerous and then grade the cancerous brain MRI images into several types of brain tumor simultaneously.

Bibliography

Book

Brain & Spine Foundation. *A Guide for Patients and Carers*. www.brainandspine.org.uk 2020, 1-56.

Cancer Council. *Understanding Brain Tumors: A Guide for People with Brain or Spinal Cord Tumors, their Families and Friends*. 2020. May 2020 edition, First published 1995, Next Ed. 2022.

Cancer Support Community. *Frankly Speaking about Brain Tumors.* www.cancersupportcommunity.org 2013, 1-96

Drevelegas, A. & Papanikolou, N. *Imaging Modalities in Brain Tumors Imaging of Brain Tumors with Histological Correlations*. Berlin: Springer; 2011; Chapter 2, 13-34.

Witten, I.H. & Frank, E. *Data Mining: Practical Machine Learning Tools and Techniques* A volume in the Morgan Kaufmann Series in Data Management Systems. Third Edition, 2011, 607-629.

Conference

Bezdan, T. Milosevic, S. K, V. Zivkovic, M. Bacanin, N. & Strumberger, I. *Optimizing Convolutional Neural Network by Hybridized Elephant Herding Optimization Algorithm for Magnetic Resonance Image Classification of Glioma Brain Tumor Grade*. Zooming Innovation in Consumer Technologies Conference (ZINC), 2021, 171–176.

Christian, S. Vincent, V. Sergey, I. Jon, S. & Zbigniew. W. *Rethinking the Inception Architecture for Computer Vision*. In Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, 2818–2826.

Ferran, P. Dario, G. & Jesus L. *Training CNNs Using High-Resolution Images of Variable Shape.* 6th BSC Severo Ochoa Doctoral Symposium. 2023, 60-61.

Gokila Brindha, P. Kavinraj, M. Manivasakam, P. & Prasanth P. *Brain Tumor Detection from MRI Images Using Deep Learning Techniques.* 2021 IOP Conference Series: Materials Science and Engineering, 1055, 2021, 012115.

Kavita, B. Varun, R. Sanjay, S. & Vijay, S. *Brain Tumor Detection Using Deep Learning Techniques.* Proceedings of the 4th International Conference on Advances in Science & Technology (ICAST2021), 2021, 1-5.

Lina, C. & Shahram, L. Department of Electrical and Computer Engineering University of Nevada, Las Vegas (UNLV). *Machine Learning and Deep Learning Techniques to Predict Overall Survival of Brain Tumor Patients using MRI.* 2017 IEEE 17th International Conference on Bioinformatics and

Bioengineering.

Mingxing, T. & Quoc, L. *Efficientnet*: *Rethinking Model Scaling for Convolutional Neural Networks*. In International Conference on Machine Learning, 2019, 6105–6114.

Mohammed, M. Khan, M.B. & Bashier, M.B.E. *Machine Learning: Algorithms and Applications*. CRC Press, 2016.

Ostrom, Q.T. & Barnholtz-Sloan, J.S. *Current State of Our Knowledge on Brain Tumor Epidemiology*. Current Neurology and Neuroscience Reports 11: 2011, 329-335. Link: https://bit.ly/2T5lI7V.

Pashaei, A. Sajedi, H. & Jazayeri, N. *Brain Tumor Classification via Convolutional Neural Network and Extreme Learning Machines*. 2018 8th International Conference on Computer and Knowledge Engineering (ICCKE), 2018 314–319.

Paul, J. S. Plassard, A. J. Landman, B. A. & Fabbri, D. *Deep Learning for Brain Tumor Classification*. Proceedings of the SPIE, Volume 10137, id. 1013710 16 pp. 2017.

Pinckaers, J.H.F.M. & Litjens, G.J.S. *Training Convolutional Neural Networks with Megapixel Images*. 1st international conference on Medical Imaging with Deep Learning (MIDL 2018), Amsterdam. 2018, 1-3.

Tawsin, U.A. Mohammad, N.J. Mohammad, S.H. Karl, A. & Mohammed, S.H. *An Integrated Real-Time Deep Learning and Belief Rule Base Intelligent System to Assess Facial Expression under Uncertainty***.** Researchgate Conference Paper · June 2020, 1-7.

Zahra, S. Safiyeh, R. Alireza, N. Mehdi, A. Hamidreza, Z. Nader, K. Ali, E. & Shadrokh, S. *Brain Tumor Segmentation Using Deep Learning by Type Specific Sorting of Images*. International conference on Computer Vision and Pattern Recognition (CVPR), 20th Sep 2018.

Internet Resources

Abhijeet, P. *Concept of AlexNet:- Convolutional Neural Network*. Analytics Vidhya, 2020, https://medium.com/analytics-vidhya/concept-of-alexnet-convolutional-neural-network-6e73b4f9ee30.

Ahmed, H. *Br35H :: Brain Tumor Detection 2020.* 2021, https://www.kaggle.com/datasets/ahmedhamada0/brain-tumor-detection.

Ajitesh, K. *Different Types of CNN Architectures Explained: Examples*. Analytics Yogi. 2023, https://vitalflux.com/different-types-of-cnn-architectures-explained-examples/.

Amapiano. *Introduction to Machine Learning 2023: PDF Download. A-Brief-Introduction-to-Machine-Learning-for-Engineers*. 2023, https://www.citimuzik.com/2023/03/introduction-to-machine-learning-2023-pdf-download.html.

Amazon Web Services. *What is an IDE (Integrated Development Environment)?* 2024, https://aws.amazon.com/what-is/ide/.

Anish, S.W. *Types of Optimization Algorithms used in Neural Networks and Ways to Optimize Gradient Descent*. Nerd for Tech. 2021, https://medium.com/nerd-for-tech/types-of-optimization-algorithms-used-in-neural-networks-and-ways-to-optimize-gradient-descent-1e32cdcbcf6c.

Artem, O. *What Is Deep Learning and How Does It Work?* 2022, https://builtin.com/authors/artem-oppermann.

Ayush, G. *A Comprehensive Guide on Optimizers in Deep Learning*. Analytics Vidhya. 2023, https://www.analyticsvidhya.com/blog/2021/10/a-comprehensive-guide-on-deep-learning-optimizers/.

Jason, B. *What is Deep Learning?* 2020, https://machinelearningmastery.com/what-is-deep-learning/.

Chinmay, B. *A Practical Guide to Selecting CNN Architectures for Computer Vision Applications from LeNet to EfficientNet: Choosing the Best CNN Architecture for Your Project.* 2023, https://levelup.gitconnected.com/a-practical-guide-to-selecting-cnn-architectures-for-computer-vision-applications-4a07ef90234.

Coursera. *What Is Deep Learning? Definition, Examples, and Careers*. 2023, https://www.coursera.org/articles/what-is-deep-learning.

Derrick, M. *Image Classification with Convolutional Neural Networks (CNNs).* KDnuggets. 2022, https://www.kdnuggets.com/2022/05/image-classification-convolutional-neural-networks-cnns.html.

Ed, B. & Kate, B. *Definition deep learning*. Techtarget. 2021, https://www.techtarget.com/searchenterpriseai/definition/deep-learning-deep-neural-network.

Eric, M. *Image Classification with Variable Input Resolution in Keras***.** Mindboard. Mar 20, 2020. Website, https://medium.com/mindboard/image-classification-with-variable-input-resolution-in-keras-cbfbe576126f.

Gaudenz, B. *A Complete Guide to Image Classification in 2023.* 2023, https://viso.ai/computer-vision/image-classification/.

geeksforgeeks. *Difference between AlexNet and GoogleNet*. 2023, https://www.geeksforgeeks.org/difference-between-alexnet-and-googlenet/.

geeksforgeeks. *OpenCV Tutorial in Python*. 2024, https://www.geeksforgeeks.org/opencv-python-tutorial/?ref=shm.

geeksforgeeks. *Python Numpy*. 2023, https://www.geeksforgeeks.org/python-numpy/.

Intelerad. *What is DICOM Image Format & Why is It Important in Radiology?* 2023, https://www.intelerad.com/en/2023/02/23/handling-dicom-medical-imaging-data/.

Jarkko, S. *Technologies for Learning and Development*. 2023, https://sites.uef.fi/edtech/research-labs-and-focus-groups/ict-for-development/.

Jean, D. *Brain MRI Images for Brain Tumor Detection*. 2020, https://www.kaggle.com/datasets/jjprotube/brain-mri-images-for-brain-tumor-detection.

Jeremy, H. F. *Progressive Resizing.* 2018, https://www.fast.ai/2018/04/30/dawnbench-fastai/.

Jonas, C. *7 Best Image Classification Models You Should Know in 2023*. 2023, https://jonascleveland.com/best-image-classification-models/#:~:text=Some%20of%20the%20most%20popular,resources%20for%20training%20and%20inference.

Larson, J.B. *Python Image Processing: A Tutorial*. 2023, https://builtin.com/.

Masoud, N. *Brain Tumor MRI Dataset.* 2021, https://www.kaggle.com/datasets/masoudnickparvar/brain-tumor-mri-dataset.

MathWorks. *What Is Deep Learning? Apply deep learning to the design of smart engineering systems*. 2023, https://www.mathworks.com/discovery/deep-learning.html.

Mikhael, A. P. G. *Br35H :: Brain Tumor Detection 2020.* 2022, https://www.kaggle.com/datasets/mikhaelapg/brain-tumor-detection.

Mohammad, H. H. · *Crystal Clean: Brain Tumors MRI Dataset.* 2023, https://www.kaggle.com/datasets/mohammadhossein77/brain-tumors-dataset.

Nagesh, S.C. *Optimization Algorithms in Neural Networks*. KDnuggets. 2020, https://www.kdnuggets.com/2020/12/optimization-algorithms-neural-networks.html.

Navoneel, C. *Brain MRI Images for Brain Tumor Detection.* 2019, https://www.kaggle.com/datasets/navoneel/brain-mri-images-for-brain-tumor-detection?select=yes.

Neetika-khandelwal. *Image Processing in Python: Algorithms, Tools, and Methods You Should Know*. MLOps Blog. 2023, https://neptune.ai/blog/.

Numpy. What is NumPy? 2022, https://numpy.org/doc/stable/user/whatisnumpy.html#what-is-numpy.

Open Access Publisher. *Brain Tumor.* 2022, https://www.omicsonline.org/open-access-journals-list.php.

Panagiotis, A. *How to Handle Large Images to Train CNNs?* 2023, https://www.baeldung.com/cs/.

Pratham, G. B*rain_Tumor_Classification*. 2023, https://www.kaggle.com/datasets/prathamgrover/brain-tumor-classification.

Quora. Why is quantitative research important to an IT specialist? 2023, https://www.quora.com/Why-is-quantitative-research-important-to-an-IT-specialist.

RedHat. What is an IDE? 2019, https://www.redhat.com/en/topics/middleware/what-is-ide.

Rupam. *What are Machine Learning Applications? Top 10 Industry and Real-World Use Cases.* 2022, https://emeritus.org/blog/machine-learning-what-are-machine-learning-applications/.

Sanket, D. *Various Optimization Algorithms for Training Neural Network. Towards Data Science.* 2019, https://towardsdatascience.com/optimizers-for-training-neural-network-59450d71caf6.

Saumyasaxena2730. *Introduction to Deep Learning*. 2023, https://www.geeksforgeeks.org/introduction-deep-learning/.

Shipra, S. *Introduction to the Architecture of Alexnet.* 2023, https://www.analyticsvidhya.com/blog/2021/03/introduction-to-the-architecture-of-alexnet/.

Shreya, G. *Brain MRI Scans for brain tumor classification.* 2023, https://www.kaggle.com/datasets/shreyag1103/brain-mri-scans-for-brain-tumor-classification.

Siddhesh, B. *AlexNet Architecture Explained*. 2022, https://medium.com/@siddheshb008/alexnet-architecture-explained-b6240c528bd5.

Silverio, M. *What's So Great About Jupyter Notebook?* 2023, https://builtin.com/software-engineering-perspectives/jupyter-notebook.

Singh, V. *Difference Between Jupyter Notebook and Python IDLE*. 2023, https://www.shiksha.com/online-courses/articles/jupyter-notebook-vs-python-idle/#:~:text=Jupyter%20Notebook%20is%20an%20interactive,writing%20and%20executing%20Python%20code.

Sohaib, Zafar. *Choosing the Right Pre-Trained Model: A Guide to VGGNet, ResNet, GoogleNet, AlexNet, and Inception.* 2023, https://medium.com/@sohaib.zafar522/choosing-the-right-pre-trained-model-a-guide-to-vggnet-resnet-googlenet-alexnet-and-inception-db7a8c918510#:~:text=AlexNet%20and%20ResNet%2D152%20both,more%20training%20time%20and%20energy.

Sorcha, S. *Image Classification in AI: How it works.* 2022, https://levity.ai/blog/image-classification-in-ai-how-it-works.

Steve, C. *Python Tutorial for Beginners: Learn Programming Basics [PDF] Guru99 2022.* 2022, https://www.guru99.com/.

SuperAnnotate. *What Is Image Classification? Basics You Need To Know.* 2023, https://www.superannotate.com/blog/image-classification-basics.

Techjury. *How Much Data Is Created Every Day in 2022?.* 2022, https://techjury.net/blog/how-much-data-is-created-every-day/#gref.

Zubair, H. *Best CNN Architecture For Image Processing.* 2022, https://www.folio3.ai/blog/best-cnn-architecture-for-image-processing/.

Journal

Abd-ellah, M. K. Awad, A. I. Ashraf, A. M. & Hamed, H. F. A. *A Review on Brain Tumor Diagnosis from MRI Images: Practical Implications, Key Achievements, and Lessons Learned.* **Magnetic Resonance Imaging**, 61, 2019, 300–318.

Abd-Ellah, M. K. Awad, A. I. Khalaf, A. A. M. & A., H. F. *Two-Phase Multi-Model Automatic Brain Tumour Diagnosis System from Magnetic Resonance Images Using Convolutional Neural Networks*. **EURASIP Journal on Image and Video Processing**, 97, 2018.

Abhishek, A. Aditya, R. Aravind, S. Naveen C.R. & Deepa P.L. *Brain Tumor Detection from Brain MRI using Deep Learning*. **International Journal of Innovative Research in Applied Sciences and Engineering (IJIRASE)** 3(2), 2019, 458-465

Abolanle, K.A.A. Amina, S. Muhammad, A. Hina, A. Omowumi, K.T. *Brain Tumor: An Overview of the Basic Clinical Manifestations and Treatment.* **Glob J Cancer Ther** 6(1), 2020, 038-041.

Ali, I. Cem, D. & Melike Ş. *Review of MRI-based Brain Tumor Image Segmentation Using Deep Learning Methods*. **Procedia Computer Science** 102 (2016), 2016, 317 – 324.

Ali, M.A. Hiam, A. Isam, A.Q. Amin, A. & Wafaa, A. *Brain Tumor Classification Using Deep Learning Technique - A Comparison between Cropped, Uncropped, and Segmented Lesion Images with Different Sizes*. **International Journal of Advanced Trends in Computer Science and Engineering,** 8(6), 2019, 3684–3691.

Amin, J. Sharif, M. Gul, N. Raza, M. Anjum, M. A. Nisar, M. W. & Bukhari, S. A. C. *Brain Tumor Detection by Using Stacked Autoencoders in Deep Learning*. **Journal of Medical Systems**. 44(2), 2020.

Anubhav, K.P. Dileep, K.M, Víctor, D.J.M. Mohan, B.R. & Achyutha, P. *Machine Learning Approach for Prediction of the Online User Intention for a Product Purchase*. **International Journal on Recent and Innovation Trends in Computing and Communication,** 11(1s), 2023, 41-51.

Athency, A, Ancy, B.M.A. Fathima, K.A. Dilin, R. & Binish, M.C. *"Brain Tumor Detection and Classification in MRI"* **International Journal of Innovative Research in Science, Engineering and Technology**. 6(5), 32, 2017.

Cao, L. *Data Science: A Comprehensive Overview*. **ACM ComputSurv (CSUR)**. 50(3), 2017, 43.

Chen, Y.N. Fan, K.C. Chang, Y.L. & Moriyama, T. *Special Issue Review: Artificial Intelligence and Machine Learning Applications in Remote Sensing*. **Remote Sens.** 15(569), 2023, Page 1-10.

Chinmayi, P. Akilandeswari, L. Prabu Kumar, M. &. Murali Babu, K. *An Efficient Deep Learning Neural Network Based Brain Tumor Detection System*" **International Journal of Pure and Applied Mathematics.** 117(17), 2017, 151-160.

Ehret, F. Kaul, D. Clusmann, H. Delev, D. & Kernbach, J. M. *Machine Learning-Based Radiomics in Neuro-Oncology*. **Machine Learning in Clinical Neuroscience,** 134, 2022, 139–151.

Elad, H. Berry, W. Itay, H. Tal, B. Torsten, H. & Daniel, S. *Mix & Match: Training Convnets with Mixed Image Sizes for Improved Accuracy, Speed and Scale Resiliency*. **arXiv.** 2019, 1-13.

Ertuğrul, Ö. F. *A Novel Clustering Method Built on Random Weight Artificial Neural Networks and Differential Evolution*. **Soft Computing**, 24, 2020, 12067–12078.

Eun, Y. O. & Donggil, S. *Developmental Research on An Interactive Application for Language Speaking Practice Using Speech Recognition Technology*. **Educational Technology Research and Development.** 69, 2021, 861–884.

Eyal, L. Rajan, J. Narges, R. Girish, M. F. & Yvonne, W. L. *State of the art: Machine learning applications in glioma imaging*. **American Journal of Roentgenology**, 212(1), 2019, 26-37.

Faisal, M. Shabir, A. & Taeg, K.W. *An Efficient Optimization Technique for Training Deep Neural Networks.* **Mathematics.** 11(6), 2023, 1360.

Fanjie, K. & Ricardo, H. *Efficient Classification of Very Large Images with Tiny Objects*. Computer Vision Foundation (CVF) Open Access. 2022, 2384-2394.

Wessam M. S. & Ahmed, S. *A Novel Framework for Brain Tumor Detection Based on Convolutional Variational Generative Models*. Multimedia Tools and Applications, 81, 2022, 16441–16454.

Gaur, L. Bhandari, M. Razdan, T. Mallik, S. & Zhao, Z. *Explanation-Driven Deep Learning Model for Prediction of Brain Tumour Status Using MRI Image Data*. **Front. Genet.** 13:822666. 2022.

Gao, X. W. Hui, R. & Tian, Z. *Classification of CT Brain Images Based on Deep Learning Networks.* **Computer Methods and Programs in Biomedicine**, *138*, 2017, 49–56.

Ghassemi, N. Shoeibi, A. & Rouhani, M. *Deep Neural Network with Generative Adversarial Networks Pre-Training for Brain Tumor Classification Based on MR Images*. **Biomedical Signal Processing and Control**, 57, 2020, 101678.

Grauman, K. & Darre ll, T. *The Pyramid Match Kernel: Discriminative Classification with Sets of Image Features*, in ICCV, 2005.

Han, J. Pei, J. & Kamber, M. *Data Mining: Concepts and Techniques*. **Amsterdam: Elsevier,** 2011.

He, K. Zhang, X. Ren, S. & Sun, J. *Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition.* **[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence**, 37(9), 2015, 1904-1916.

Heba, M. El-Sayed, A. E. El-Sayed, M.E. & Abdel-Badeeh, M. S. *Classification Using Deep Learning Neural Networks for Brain Tumors*" **Future Computing and Informatics Journal**. 3, 2018, 68-71.

Ismael, S. A. A. Mohammed, A. & Hefny, H. *An Enhanced Deep Learning Approach for Brain Cancer MRI Images Classification Using Residual Networks.* **Artificial Intelligence in Medicine**, 102, 2020, 101779.

Karen, S. & Andrew, Z. *Very Deep Convolutional Networks for Large-Scale Image Recognition.* **arXiv preprint** arXiv:1409.1556, 2014.

Ke, Q. Zhang, J. Wei, W. Damasevicius, R. & Wozniak, M. *Adaptive Independent Subspace Analysis of Brain Magnetic Resonance Imaging Data*. **IEEE Access**. 7, 2019, 12252–12261.

Khawaldeh, S. Pervaiz, U. Rafiq, A. & Alkhawaldeh, R.S. *Noninvasive Grading Of Glioma Tumor Using Magnetic Resonance Imaging with Convolutional Neural Networks.* **Applied Sciences**, 8(1), 2018, 27.

Laith, A. Jinglan, Z. Amjad, J.H. Ayad, A. Ye, D. Omran, A. Santamaría, J. Mohammed A.F. Muthana, A. & Laith, F. *Review of Deep Learning: Concepts, CNN Architectures, Challenges, Applications, Future Directions*. **Journal of Big Data**. 2021, 1–74.

Lazebnik, S. Schmid, C. & Ponce, J. *Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories*. In **CVPR**, 2006.

LeCun, Y. Bengio, Y. & Hinton, G. *Deep Learning*. **Nature**. 521(7553), 2015, 436–44.

Lin, M. Chen, Q. & Yan, S. Network in Network. **Computer Science**, 2013.

Litjens, G. Kooi, T. Bejnordi, B. E. Setio, A. A. A. Ciompi, F. Ghafoorian, M. Laak, J. A. W. M. Ginneken, B. & Sánchez, C. I. *A Survey on Deep Learning in Medical Image Analysis*. **Medical Image Analysis**, 42, 2017, 60–88.

Maxwell, A.E. Warner, T.A. & Guillen, L.A. *Accuracy Assessment in Convolutional Neural Network-Based Deep Learning Remote Sensing Studies—Part 1: Literature Review.* **Remote Sens.** 13, 2021, 2450.

McCallum, A. *Information Extraction: Distilling Structured Data from Unstructured Text*. **Queue**. 3(9), 2005, 48–57.

Moez, K. *Convolutional Neural Networks: A Survey*. **Computers.** 12(8), 2023, 151.

Muhammad, I. S. Muhammad, A. K. Musaed, A. Khursheed, A. & Mudassar, R. *A decision support system for multimodal brain tumor classification using deep learning.* **Complex & Intelligent Systems**, 2021, 1-14.

Neha, M. Yogesh, K.M. Shruti, M. & Divya, M. *Detection of Brain Tumor in MRI Image through Fuzzy-Based Approach*. High-Resolution Neuroimaging-Basic Physical Principles and Clinical Applications, IntechOpen, 2018.

Noreen, N. Palaniappan, S. Qayyum, A. Ahmad, I. & Shoaib, M. I. M. *A Deep Learning Model Based on Concatenation Approach for the Diagnosis of Brain Tumor.* **IEEE Access.** 8, 2020, 55135–55144.

Özyurt, F. Sert, E. Avci, E. & Dogantekin, E. *Brain Tumor Detection Based on Convolutional Neural Network with Neutrosophic Expert Maximum Fuzzy Sure Entropy*. **Measurement**, 147, 2019.

Raj, R. J. S. Shobana, S. J. Pustokhina, I. V. Pustokhin, D. A. Gupta, D. & Shankar, K. *Optimal Feature Selection-Based Medical Image Classification Using Deep Learning Model in Internet of Medical Things*. **IEEE Access**, *8*, 2020, 58006–58017.

Raymond, A.R. Developmental Research for Information and technology. Studoc. 2023.

Rehman, A. Khan, M.A. Saba, T. Mehmood, Z. Tariq, U. & Ayesha, N. *Microscopic brain tumor detection and classification using 3D CNN and feature selection architecture*. **Microsc Res Tech**, 84(1), 2020, 133–149.

Rehman, A. Naz, S. Razzak, M. I. Akram, F. & Imran, M. *A Deep Learning-Based Framework for Automatic Brain Tumors Classification Using Transfer Learning*. **Circuits, Systems, and Signal Processing.** 39, 2020. 757–775.

Rehman, H.U. Masood, S. & Khan, M.A. *Brain Tumor Classification using Deep Learning Methods.* **International Journal of Information Systems and Computer Technologies (IJISCT)**, Vol. 1(1), 2022, 37-53.

Russakovsky, O. Deng, J. Su, H. Krause, J. Satheesh, S. Ma, S. Huang, Z. Karpathy, A. Khosla, A. Bernstein, M. Berg, A.C. Fei-Fei L. *Imagenet Large Scale Visual Recognition Challenge.* **arXiv:1409.0575 [cs.CV]**, 3, 2015, 1-43.

Sarker, I.H. Deep Cybersecurity: *A Comprehensive Overview from Neural Network and Deep Learning Perspective*. **SN Comput Sci**. 2021.

Sarker, I.H. Hoque, M.M. MdK, U. & Tawfeeq A. *Mobile Data Science and Intelligent Apps: Concepts, AI-Based Modeling and Research Directions*. **Mob Netw Appl.** 2020, 1–19.

Sarker, I.H. *Machine Learning: Algorithms, Real World Applications and Research Directions*. **A SPRINGER NATURE Journal. SN Computer Science.** 2(160), 2021, 1-21.

Sarker, I.H. Watters, P. & Kayes, A.S.M. *Effectiveness Analysis of Machine Learning Classification Models for Predicting Personalized Context-Aware Smartphone Usage*. **J Big Data**. 6(1), 2019, 1–28.

Saurabh, K. Iram, A. Shubhi, G. Anand, K.S. & Vivek, J. *Brain Tumor Detection Using Image Processing*. **International Journal of Information Sciences and Application (IJISA),** 11(1), 2019, 20 – 25.

Sajjad, M. Khan, S. Muhammad, K. Wu, W. Ullah, A. & Baik, S.W. *Multi-Grade Brain Tumor Classification Using Deep CNN with Extensive Data Augmentation*. **Journal of Computational Science.** 30, 2019, 174-82. 2019.

Seetha, J. & Raja S.S. *Brain Tumor Classification using Convolutional Neural Networks.* **Biomed. & Pharmacol. J** 11(3), 2018, 1457-1461.

Shamsul, H. John, Y. Herbert, F.J. Mohammad, M.H. Giancarlo, F. & Michael, B. *A Hybrid Feature Selection with Ensemble Classification for Imbalanced Healthcare Data: A Case Study for Brain Tumor Diagnosis.* ***IEEE Access***, 4, 2017, 9145 - 9154.

Sharma, k. Kaur, A. & Gujral, S. *Brain Tumor Detection Based on Machine Learning Algorithms*. **International Journal of Computer Applications (0975 – 8887),** (103)1, 2014, 7-11.

Shree, N. V. & Kumar, T. N. R. *Identification and Classification of Brain Tumor MRI with Feature Extraction Using DWT and Probabilistic Neural Network*. **Brain Infomatic.** 5(1), 2018, 23-30.

Shrestha, A. & Mahmood, A. *Review of Deep Learning Algorithms and Architectures*. **IEEE Access**.7, 2019, 53040–65.

Sung, H. Ferlay, J. Siegel, R. L. Laversanne, M. Soerjomataram, I. & Jemal, A. *Global Cancer Statistics 2020: GLOBOCAN Estimates of Incidence and Mortality Worldwide for 36 Cancers in 185 Countries.* **Cancer J. Clin.** 71, 2021, 209–249.

Tero, K. Timo, A. Samuli, L. & Jaakko, L. *Progressive Growing of Gans for Improved Quality, Stability, and Variation***. arXiv preprint** arXiv:1710.10196, 2017.

Touvron, H. Andrea, V. Matthijs, D. & Herv´e, J. *Fixing the Train-Test Resolution Discrepancy*. **CoRR**, abs/1906.06423, 2019.

Vidya, M.M. *Covid-19 Prediction using Machine Learning*. **International Research Journal of Modernization in Engineering Technology and Science**. 05(05), 2023. 1180-1185.

Volodymyr, M. Koray, K. David, S. Andrei, A.R. Joel, V, Marc, G.B. Alex, G. Martin, R. Andreas, K.F. Georg, O. Stig, P. Charles, B. Amir, S. Ioannis, A. Helen, K. Dharshan, K. Daan, W. Shane, L. & Demis, H. . *Human-Level Control Through Deep Reinforcement Learning*. **Nature**. 518, 2015. 529–533.

Wang, Y. Wei, G.Y. & Brooks, D. *Benchmarking TPU, GPU, and CPU Platforms for Deep Learning*. **arXiv.** 1907.10701. 2019**.**

Widhiarso, W. Yohannes, Y. & Cendy, P. *Brain Tumor Classification Using Gray Level Co-occurrence Matrix and Convolutional Neural Network.* **IJEIS (Indonesian Journal of Electronics and Instrumentation Systems).** 1 8, 2018, 179-190.

Wojciech, S. Thomas, W. and Robert, M. K. *Explainable Artificial Intelligence: Understanding, Visualizing and Interpreting Deep Learning Models*. **arXiv:1708.08296 [cs.AI],** 2017.

Yanping, H. Yonglong, C. Dehao, C. HyoukJoong, L. Jiquan, N. Quoc, V. L. & Zhifeng, C. *Gpipe: Efficient Training of Giant Neural Networks Using Pipeline Parallelism*. **arXiv preprint** arXiv:1811.06965, 2018.

Yili, Q. Yaobin, K. & Wei, Y. *A Solution for Input Limit in CNN Due to Fully Connected Layer.* **IEEE Xplore.** 2018, 1-6.

Zhang, Z. Cui, P. & Zhu, *W. Deep Learning on Graphs: A Survey.* **IEEE Trans Knowl Data Eng.** 29813, 33, 2020.

Appendix

Source code

Import Necessary Libraries for the Project

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import os, shutil
import cv2
import matplotlib.image as mpimg
import seaborn as sns
%matplotlib inline
plt.style.use('ggplot')
import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Flatten, Dense, Dropout
from tensorflow.keras.optimizers import SGD, Adam
from tensorflow.keras.callbacks import ModelCheckpoint, EarlyStopping, ReduceLRO
nPlateau
```

Unzip the Zip file Downloaded from Kaggle

```
# Dataset
import zipfile
z = zipfile.ZipFile('archive.zip')
z.extractall()
```

Rename All the Images for Uniformity Sake

```
folder = 'brain_tumor_dataset/yes/'
count = 1
for filename in os.listdir(folder):
  source = folder + filename
  destination = folder + "Y_" +str(count)+".jpg"
  os.rename(source, destination)
  count += 1
print("All files are renamed in the yes dir.")
```

All files are renamed in the yes dir.

```
folder = 'brain_tumor_dataset/no/'
count = 1
for filename in os.listdir(folder):
  source = folder + filename
  destination = folder + "N_" +str(count)+".jpg"
  #os.rename(source, destination)
  os.replace(source, destination)
  count += 1
print("All files are renamed in the no dir.")
```

List the Number of Images in the Dataset

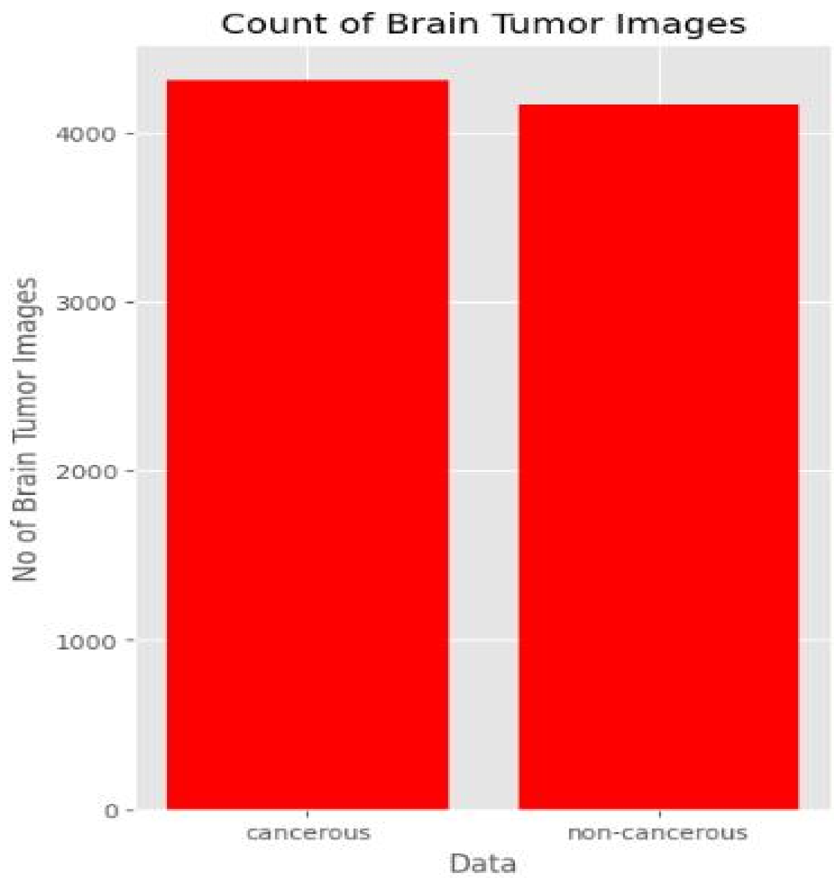
```
listyes = os.listdir("brain_tumor_dataset/yes/")
number_files_yes = len(listyes)
print("Number of Yes file: ", number_files_yes)
listno = os.listdir("brain_tumor_dataset/no/")
number_files_no = len(listno)
print("Number No file:     ", number_files_no)
```

Number of Yes file: 4304

Number No file: 4165

Histogram showing the number of Tumorous and Non-Tumorous images

```
data = {'tumorous': number_files_yes, 'non-tumorous': number_files_no}
typex = data.keys()
values = data.values()
fig = plt.figure(figsize=(5,7))
plt.bar(typex, values, color="red")
plt.xlabel("Data")
plt.ylabel("No of Brain Tumor Images")
plt.title("Count of Brain Tumor Images")
plt.show()
```

Function to Calculate the Execution Time

```
def timing(sec_elapsed):
  h = int(sec_elapsed / (60*60))
  m = int(sec_elapsed % (60*60) / 60)
  s = sec_elapsed % 60
  return f"{h}:{m}:{s}"
```

Function to Augment the Data

```
def augmented_data(file_dir, n_generated_samples, save_to_dir):
  data_gen = ImageDataGenerator(rotation_range=10,
           width_shift_range=0.1,
           height_shift_range=0.1,
           shear_range=0.1,
           brightness_range=(0.3, 1.0),
           horizontal_flip=True,
           vertical_flip=True,
           fill_mode='nearest')
  for filename in os.listdir(file_dir):
    image = cv2.imread(file_dir + '/' + filename)
    image = image.reshape((1,) + image.shape)
    save_prefix = 'aug_' + filename[:-4]
    i = 0
    for batch in data_gen.flow(x = image, batch_size = 1, save_to_dir = save_to_dir, s
ave_prefix = save_prefix, save_format = "jpg"):
      i += 1
```

```
    if i > n_generated_samples:
        break
```

Augmenting the Data

```
import time
start_time = time.time()
yes_path = 'brain_tumor_dataset/yes'
no_path = 'brain_tumor_dataset/no'

augmented_data_path = 'augmented_data/'
augmented_data(file_dir = yes_path, n_generated_samples=4, save_to_dir=augmented_data_path+'yes')
augmented_data(file_dir = no_path, n_generated_samples=4, save_to_dir=augmented_data_path+'no')
end_time = time.time()
execution_time = end_time - start_time
print(timing(execution_time))
1:2:16.752981185913086
```

Calculate the Percentage of Tumorous and Non-Tumorous Images in the Dataset

```
def data_summary(main_path):
  yes_path = "augmented_data/yes/"
  no_path = "augmented_data/no/"
  n_pos = len(os.listdir(yes_path))
  n_neg = len(os.listdir(no_path))
    n = (n_pos + n_neg)
    pos_per = (n_pos* 100)/n
  neg_per = (n_neg* 100)/n
```

```
    print(f"Number of sample: {n}")
    print(f"{n_pos} Number of Tumor sample in percentage: {pos_per}%")
    print(f"{n_neg} Number of Non TUMOR sample in percentage: {neg_per}%")
augmented_data_path = 'augmented_data/'
data_summary(augmented_data_path)
Number of sample: 42337
21517 Number of Tumor sample in percentage: 50.82315704938942%
20820 Number of Non TUMOR sample in percentage: 49.17684295061058%
```

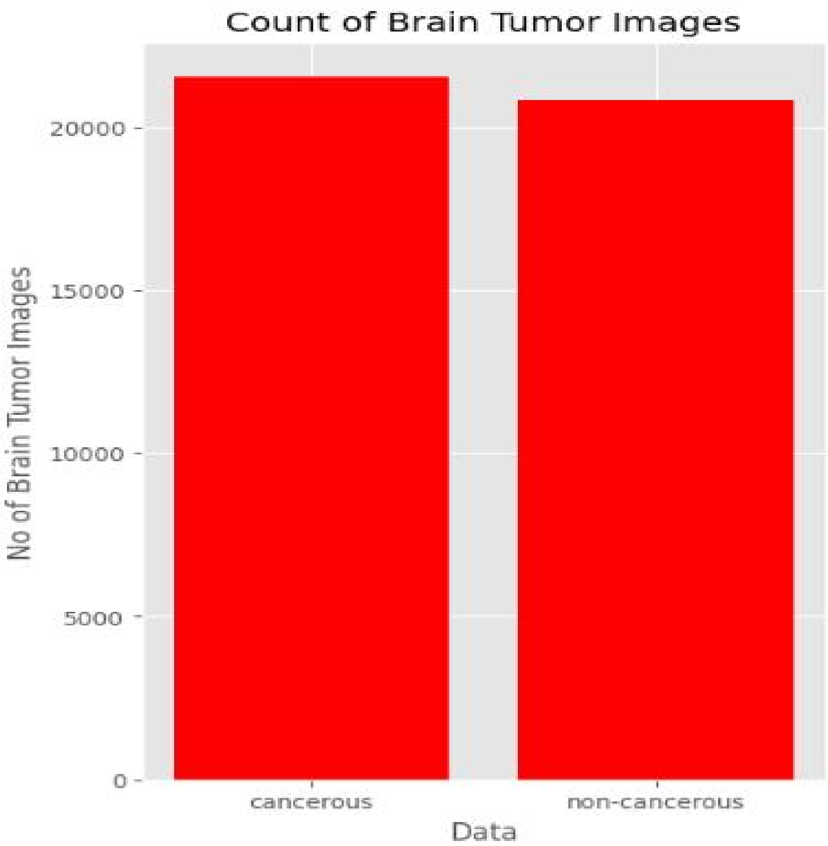
Listing Number of Images after Augmentation

```
listyes = os.listdir("augmented_data/yes/")
number_files_yes = len(listyes)
print(" Number of Tumor images after augumentation: ", number_files_yes)
listno = os.listdir("augmented_data/no/")
number_files_no = len(listno)
print(" Number of Non Tumor images after augumentation", number_files_no)
 Number of Tumor images after augumentation: 21517
 Number of Non Tumor images after augumentation 20820
```

Histogram Showing the Number of Tumorous and Non-Tumorous Images in the Dataset after Augmentation

```
data = {'tumorous': number_files_yes, 'non-tumorous': number_files_no}
typex = data.keys()
values = data.values()
fig = plt.figure(figsize=(5,7))
```

```
plt.bar(typex, values, color="red")
plt.xlabel("Data")
plt.ylabel("No of Brain Tumor Images")
plt.title("Count of Brain Tumor Images")
plt.show()
```



Function To Crop the Unwanted Black Edge of the Images

```
import imutils
def crop_brain_tumor(image, plot=False):
  gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
  gray = cv2.GaussianBlur(gray, (5,5), 0)
  thres = cv2.threshold(gray, 45, 255, cv2.THRESH_BINARY)[1]
  thres =cv2.erode(thres, None, iterations = 2)
  thres = cv2.dilate(thres, None, iterations = 2)
```

```
cnts = cv2.findContours(thres.copy(), cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
cnts = imutils.grab_contours(cnts)
c = max(cnts, key = cv2.contourArea)
extLeft = tuple(c[c[:,:,0].argmin()][0])
extRight = tuple(c[c[:,:,0].argmax()][0])
extTop = tuple(c[c[:,:,1].argmin()][0])
extBot = tuple(c[c[:,:,1].argmax()][0])
new_image = image[extTop[1]:extBot[1], extLeft[0]:extRight[0]]
if plot:
    plt.figure()
    plt.subplot(1, 2, 1)
    plt.imshow(image)
    plt.tick_params(axis='both', which='both',
        top=False, bottom=False, left=False, right=False,
        labelbottom=False, labeltop=False, labelleft=False, labelright=False)
    plt.title('Original Image')
    plt.subplot(1, 2, 2)
    plt.imshow(new_image)
    plt.tick_params(axis='both', which='both',
        top=False, bottom=False, left=False, right=False,
        labelbottom=False, labeltop=False, labelleft=False, labelright=False)
    plt.title('Cropped Image')
    plt.show()
return new_image
```

```
folder1 = 'augmented_data/no/'
folder2 = 'augmented_data/yes/'
for filename in os.listdir(folder1):
  img = cv2.imread(folder1 + filename)
  img = crop_brain_tumor(img, False)
  cv2.imwrite(folder1 + filename, img)
for filename in os.listdir(folder2):
  img = cv2.imread(folder2 + filename)
  img = crop_brain_tumor(img, False)
  cv2.imwrite(folder2 + filename, img)
listyes = os.listdir("augmented_data/yes/")
number_files_yes = len(listyes)
print(" Number of Tumor images after cropping the edges: ", number_files_yes)
listno = os.listdir("augmented_data/no/")
number_files_no = len(listno)
print(" Number of Non Tumor images after cropping the edges: ", number_files_no)
 Number of Tumor images after cropping the edges:  21517
 Number of Non Tumor images after cropping the edges:  20820
Create folder
if not os.path.isdir('tumorous_and_nontumorous'):
  base_dir = 'tumorous_and_nontumorous'
  os.mkdir(base_dir)
if not os.path.isdir('tumorous_and_nontumorous/train'):
  train_dir = os.path.join(base_dir , 'train')
  os.mkdir(train_dir)
if not os.path.isdir('tumorous_and_nontumorous/test'):
```

```
  test_dir = os.path.join(base_dir , 'test')
  os.mkdir(test_dir)
if not os.path.isdir('tumorous_and_nontumorous/valid'):
  valid_dir = os.path.join(base_dir , 'valid')
  os.mkdir(valid_dir)
if not os.path.isdir('tumorous_and_nontumorous/train/tumorous'):
  infected_train_dir = os.path.join(train_dir, 'tumorous')
  os.mkdir(infected_train_dir)
if not os.path.isdir('tumorous_and_nontumorous/test/tumorous'):
  infected_test_dir = os.path.join(test_dir, 'tumorous')
  os.mkdir(infected_test_dir)
if not os.path.isdir('tumorous_and_nontumorous/valid/tumorous'):
  infected_valid_dir = os.path.join(valid_dir, 'tumorous')
  os.mkdir(infected_valid_dir)
if not os.path.isdir('tumorous_and_nontumorous/train/nontumorous'):
  healthy_train_dir = os.path.join(train_dir, 'nontumorous')
  os.mkdir(healthy_train_dir)
if not os.path.isdir('tumorous_and_nontumorous/test/nontumorous'):
  healthy_test_dir = os.path.join(test_dir, 'nontumorous')
  os.mkdir(healthy_test_dir)
if not os.path.isdir('tumorous_and_nontumorous/valid/nontumorous'):
  healthy_valid_dir = os.path.join(valid_dir, 'nontumorous')
  os.mkdir(healthy_valid_dir)
original_dataset_tumorours = os.path.join('augmented_data','yes/')
original_dataset_nontumorours = os.path.join('augmented_data','no/')
```

Partitioning the Images into 80:10:10

```
files = os.listdir('augmented_data/yes/')
fnames = []
for i in range(0,17213):
  fnames.append(files[i])
for fname in fnames:
  src = os.path.join(original_dataset_tumorours, fname)
  dst = os.path.join(infected_train_dir, fname)
  shutil.copyfile(src, dst)
files = os.listdir('augmented_data/yes/')
fnames = []
for i in range(17213,19365):
  fnames.append(files[i])
for fname in fnames:
  src = os.path.join(original_dataset_tumorours, fname)
  dst = os.path.join(infected_test_dir, fname)
  shutil.copyfile(src, dst)
files = os.listdir('augmented_data/yes/')
fnames = []
for i in range(19365,21517):
  fnames.append(files[i])
for fname in fnames:
  src = os.path.join(original_dataset_tumorours, fname)
  dst = os.path.join(infected_valid_dir, fname)
  shutil.copyfile(src, dst)
```

```
files = os.listdir('augmented_data/no/')
fnames = []
for i in range(0,17213):
  fnames.append(files[i])
for fname in fnames:
  src = os.path.join(original_dataset_nontumorours, fname)
  dst = os.path.join(healthy_train_dir, fname)
  shutil.copyfile(src, dst)
files = os.listdir('augmented_data/no/')
fnames = []
for i in range(17213,19365):
  fnames.append(files[i])
for fname in fnames:
  src = os.path.join(original_dataset_nontumorours, fname)
  dst = os.path.join(healthy_test_dir, fname)
  shutil.copyfile(src, dst)
files = os.listdir('augmented_data/no/')
fnames = []
for i in range(19365,20820):
  fnames.append(files[i])
for fname in fnames:
  src = os.path.join(original_dataset_nontumorours, fname)
  dst = os.path.join(healthy_valid_dir, fname)
  shutil.copyfile(src, dst)
train_datagen = ImageDataGenerator(rescale = 1./255,
          horizontal_flip=0.4,
```

```
                 vertical_flip=0.4,
                 rotation_range=40,
                 shear_range=0.2,
                 width_shift_range=0.4,
                 height_shift_range=0.4,
                 fill_mode='nearest')
test_data_gen = ImageDataGenerator(rescale=1.0/255)
valid_data_gen = ImageDataGenerator(rescale=1.0/255)
train_generator = train_datagen.flow_from_directory('tumorous_and_nontumorous/train/', batch_size=32, class_mode='categorical',shuffle=True, seed = 42, color_mode = 'rgb')
Found 34426 images belonging to 2 classes.
test_generator = train_datagen.flow_from_directory('tumorous_and_nontumorous/test/', batch_size=32, class_mode='categorical',shuffle=True, seed = 42, color_mode = 'rgb')
Found 4304 images belonging to 2 classes.
valid_generator = train_datagen.flow_from_directory('tumorous_and_nontumorous/valid/', batch_size=32,  class_mode='categorical',shuffle=True, seed = 42, color_mode = 'rgb')
Found 3607 images belonging to 2 classes.
class_labels = train_generator.class_indices
class_name = {value: key for (key,value) in class_labels.items()}
class_name
{0: 'nontumorous', 1: 'tumorous'}
class_labels
{'nontumorous': 0, 'tumorous': 1}
```

Building the Modified Model

```
import tensorflow as tf
from tensorflow.keras import layers, models
from keras import regularizers
#def global_average_pooling(x):
def reduce_mean(x):
  return tf.reduce_mean(x, axis=[1, 2])
def Modified_Alexnet(input_shape, num_classes):
  inputs = layers.Input(shape=input_shape)
  # Layer 1
  x = layers.Conv2D(96, 11, strides=4, padding='valid', activation='relu')(inputs)
  x = layers.MaxPooling2D(3, strides=2)(x)
  x = layers.BatchNormalization()(x)
  # Layer 2
  x = layers.Conv2D(256, 5, padding='same', activation='relu')(x)
  x = layers.MaxPooling2D(3, strides=2)(x)
  x = layers.BatchNormalization()(x)
  # Layer 3
  x = layers.Conv2D(384, 3, padding='same', activation='relu')(x)
  # Layer 4
  x = layers.Conv2D(384, 3, padding='same', activation='relu')(x)
  # Layer 5
  x = layers.Conv2D(256, 3, padding='same', activation='relu')(x)
  x = layers.MaxPooling2D(3, strides=2)(x)
  x = layers.BatchNormalization()(x)
  x = layers.Dropout(0.2)(x)
```

```
# Global Average Pooling
#x = layers.GlobalAveragePooling2D()(x)
x = reduce_mean(x)
# Fully Connected Layers (removed)
# x = layers.Flatten()(x)
# x = layers.Dense(4096, activation='relu')(x)
# x = layers.Dropout(0.5)(x)
# x = layers.Dense(4096, activation='relu')(x)
  # Regularization
x = layers.Dropout(0.5)(x)
x = layers.Dense(4096, activation='relu', kernel_regularizer=regularizers.l2(0.01))(x)
x = layers.Dropout(0.5)(x)
x = layers.Dense(4096, activation='relu', kernel_regularizer=regularizers.l2(0.01))(x)
  # Output Layer
outputs = layers.Dense(num_classes, activation='sigmoid')(x)
model = models.Model(inputs, outputs)
return model
```

Function Call of the Modified Model

```
input_shape = (None, None, 3) # Variable image size
num_classes = 2 # Number of classes
model= Modified_Alexnet(input_shape, num_classes)
model.summary()
Model: "model_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
```

==

input_4 (InputLayer)	[(None, None, None, 3)]	0
conv2d_15 (Conv2D)	(None, None, None, 96)	34944
max_pooling2d_9 (MaxPooling2D)	(None, None, None, 96)	0
batch_normalization_9 (BatchNormalization)	(None, None, None, 96)	384
conv2d_16 (Conv2D)	(None, None, None, 256)	614656
max_pooling2d_10 (MaxPooling2D)	(None, None, None, 256)	0
batch_normalization_10 (BatchNormalization)	(None, None, None, 256)	1024
conv2d_17 (Conv2D)	(None, None, None, 384)	885120
conv2d_18 (Conv2D)	(None, None, None, 384)	1327488
conv2d_19 (Conv2D)	(None, None, None, 256)	884992
max_pooling2d_11 (MaxPooling2D)	(None, None, None, 256)	0
batch_normalization_11 (BatchNormalization)	(None, None, None, 256)	1024
dropout_5 (Dropout)	(None, None, None, 256)	0
tf.math.reduce_mean_1 (TFOpLambda)	(None, 256)	0
dropout_6 (Dropout)	(None, 256)	0
dense_3 (Dense)	(None, 4096)	1052672
dropout_7 (Dropout)	(None, 4096)	0

dense_4 (Dense) (None, 4096) 16781312

dense_5 (Dense) (None, 2) 8194

===

Total params: 21591810 (82.37 MB)

Trainable params: 21590594 (82.36 MB)

Non-trainable params: 1216 (4.75 KB)

Optimizer

sgd = tf.keras.optimizers.legacy.SGD(learning_rate=0.0001, decay = 1e-6, momentum = 0.9, nesterov = True)

Compiling the Model

model.compile(loss='categorical_crossentropy', optimizer = 'sgd', metrics=['accuracy'])

Training the Model

filepath = 'model12042024.h5'

es = EarlyStopping(monitor='val_loss', verbose = 1, mode='min',patience=2)

cp = ModelCheckpoint(filepath, monitor='val_loss', verbose = 1, save_best_only=True, save_weights_only=False, mode='auto',save_freq='epoch')

lrr = ReduceLROnPlateau(monitor='val_accuracy', patience=3, verbose = 1, factor = 0.5, min_lr = 0.0001)

history = model.fit(train_generator,steps_per_epoch=500, epochs = 10, validation_data =valid_generator, callbacks=[es,cp,lrr])

Epoch 1/10

500/500 [==========================] - ETA: 0s - loss: 42.0853 - accuracy: 0.6637

Epoch 1: val_loss improved from inf to 39.08426, saving model to model12042024.h5

C:\Deep Learning Project\ImageClassification\imageclassification\Lib\site-packages\ke
ras\src\engine\training.py:3103: UserWarning: You are saving your model as an HDF5
file via `model.save()`. This file format is considered legacy. We recommend using inst
ead the native Keras format, e.g. `model.save('my_model.keras')`.
saving_api.save_model(
500/500 [=====================] - 1684s 3s/step - loss: 42.0853 - accuracy: 0.6
637 - val_loss: 39.0843 - val_accuracy: 0.4034 - lr: 0.0100
Epoch 2/10
500/500 [======================] - ETA: 0s - loss: 34.4546 - accuracy: 0.7623
Epoch 2: val_loss improved from 39.08426 to 31.47517, saving model to model120420
24.h5
500/500 [===================] - 1831s 4s/step - loss: 34.4546 - accuracy: 0.7623
- val_loss: 31.4752 - val_accuracy: 0.6260 - lr: 0.0100
Epoch 3/10
500/500 [======================] - ETA: 0s - loss: 28.2204 - accuracy: 0.8136
Epoch 3: val_loss improved from 31.47517 to 25.62425, saving model to model120420
24.h5
500/500 [==================] - 1903s 4s/step - loss: 28.2204 - accuracy: 0.8136 -
val_loss: 25.6242 - val_accuracy: 0.7605 - lr: 0.0100
Epoch 4/10
500/500 [======================] - ETA: 0s - loss: 23.1344 - accuracy: 0.8479
Epoch 4: val_loss improved from 25.62425 to 20.91765, saving model to model120420
24.h5
500/500 [=================== - 1579s 3s/step - loss: 23.1344 - accuracy: 0.8479
- val_loss: 20.9177 - val_accuracy: 0.8564 - lr: 0.0100
Epoch 5/10

```
500/500 [==============================] - ETA: 0s - loss: 18.9894 - accuracy: 0.8536
Epoch 5: val_loss improved from 20.91765 to 17.87446, saving model to model120420
24.h5
500/500 [==============================] - 1498s 3s/step - loss: 18.9894 - accuracy: 0.8536 -
val_loss: 17.8745 - val_accuracy: 0.4843 - lr: 0.0100
Epoch 6/10
500/500 [==============================] - ETA: 0s - loss: 15.5922 - accuracy: 0.8677
Epoch 6: val_loss improved from 17.87446 to 14.36259, saving model to model120420
24.h5
500/500 [==============================] - 1515s 3s/step - loss: 15.5922 - accuracy: 0.8677 -
val_loss: 14.3626 - val_accuracy: 0.7424 - lr: 0.0100
Epoch 7/10
500/500 [==============================] - ETA: 0s - loss: 12.8140 - accuracy: 0.8730
Epoch 7: val_loss improved from 14.36259 to 11.97534, saving model to model120420
24.h5
Epoch 7: ReduceLROnPlateau reducing learning rate to 0.004999999888241291.
500/500 [==============================] - 1360s 3s/step - loss: 12.8140 - accuracy: 0.8730 -
val_loss: 11.9753 - val_accuracy: 0.6989 - lr: 0.0100
Epoch 8/10
500/500 [==============================] - ETA: 0s - loss: 11.0353 - accuracy: 0.8859
Epoch 8: val_loss improved from 11.97534 to 10.64522, saving model to model120420
24.h5
500/500 [==============================] - 1393s 3s/step - loss: 11.0353 - accuracy: 0.8859 -
val_loss: 10.6452 - val_accuracy: 0.8323 - lr: 0.0050
Epoch 9/10
500/500 [==============================] - ETA: 0s - loss: 9.9946 - accuracy: 0.8984
```

Epoch 9: val_loss improved from 10.64522 to 9.62002, saving model to model12042024.h5
500/500 [====================] - 1445s 3s/step - loss: 9.9946 - accuracy: 0.8984 - val_loss: 9.6200 - val_accuracy: 0.8489 - lr: 0.0050
Epoch 10/10
500/500 [=========================] - ETA: 0s - loss: 9.0723 - accuracy: 0.8986
Epoch 10: val_loss improved from 9.62002 to 8.74151, saving model to model12042024.h5
Epoch 10: ReduceLROnPlateau reducing learning rate to 0.0024999999441206455.
500/500 [====================] - 1791s 4s/step - loss: 9.0723 - accuracy: 0.8986 - val_loss: 8.7415 - val_accuracy: 0.8508 - lr: 0.0050

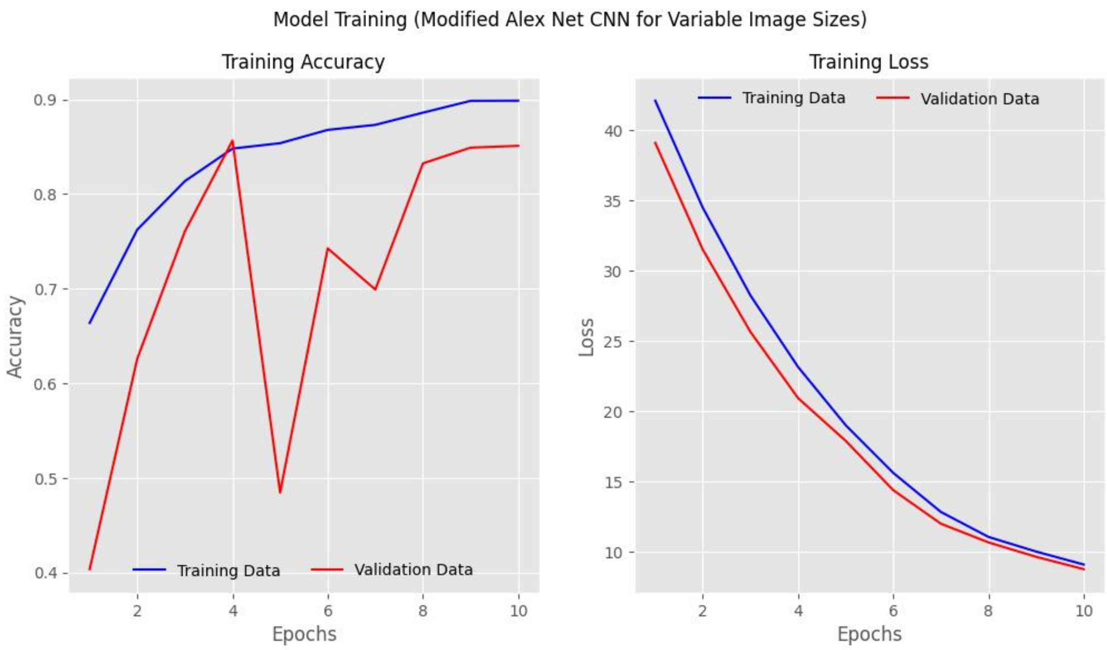
Saving the Model

model.save('model10EpochsWithNewDataSet.h5')

Plotting the Performance Graph

fig, (ax1,ax2) = plt.subplots(nrows=1, ncols=2, figsize=(12,6))
fig.suptitle("Model Training (Modified Alex Net CNN for Variable Image Sizes)", fontsize=12)
max_epoch = len(history.history['accuracy'])+1
epochs_list = list(range(1, max_epoch))
ax1.plot(epochs_list, history.history['accuracy'], color='b', linestyle='-', label='Training Data')
ax1.plot(epochs_list, history.history['val_accuracy'], color='r', linestyle='-', label='Validation Data')
ax1.set_title('Training Accuracy', fontsize=12)

```
ax1.set_xlabel('Epochs', fontsize=12)
ax1.set_ylabel('Accuracy', fontsize=12)
ax1.legend(frameon=False, loc='lower center', ncol=2)
ax2.plot(epochs_list, history.history['loss'], color='b', linestyle='-', label='Training Data')
ax2.plot(epochs_list, history.history['val_loss'], color='r', linestyle='-', label='Validation
Data')
ax2.set_title('Training Loss', fontsize=12)
ax2.set_xlabel('Epochs', fontsize=12)
ax2.set_ylabel('Loss', fontsize=12)
ax2.legend(frameon=False, loc='upper center', ncol=2)
plt.savefig("training_modified_alexnetnewdata_cnn2.jpeg", format='jpeg', dpi=100, bb
ox_inches='tight')
```

Loading the Model

model = tf.keras.models.load_model("ModifiedAlexNetModel.h5")

Evaluating the Model

 # Good Model Do far

model = tf.keras.models.load_model("ModifiedAlexNetModel.h5")

test_loss, test_acc = model.evaluate(test_generator)

#valid_loss, valid_acc = model.evaluate(valid_generator)

135/135 [==============================] - 147s 1s/step - loss: 2.1993 - accuracy: 0.8467

valid_loss, valid_acc = model.evaluate(valid_generator)

113/113 [==============================] - 127s 1s/step - loss: 2.1389 - accuracy: 0.8653

print('Test Accuracy is: {:5.2f}%'.format(100 * test_acc))

print('Validity Accuracy is: {:5.2f}%'.format(100 * valid_acc))

Test Accuracy is: 84.67%

Validity Accuracy is: 86.58%

test_label =test_generator.labels

#valid_label =valid_generator.labels

for i **in** test_generator.labels:

  print(i)

Prediction of Non-Cancerous Brain Tumor

probability_model = tf.keras.Sequential([model, tf.keras.layers.Softmax()])

predictions = probability_model.predict(test_generator)

104/104 [==============================] - 96s 918ms/step

score=predictions[3200]

```
print(
    "This image most likely belongs to {} with a {:.2f} percent confidence."
    .format(class_name[np.argmax(score)], 100 * np.max(score))
)
```

This image most likely belongs to nontumorous with a 73.10 percent confidence.

Prediction of Cancerous Brain Tumor

```
probability_model = tf.keras.Sequential([model,
                                         tf.keras.layers.Softmax()])
#predictions_on_valid_data = probability_model.predict(valid_generator, batch_size=32, verbose=0)
predictions= probability_model.predict(test_generator, batch_size=32, verbose=0)
Predict_single_image= predictions[10]
#Predict_single_image_on_valid_data = predictions_on_valid_data[400]
#Predict = predictions[400]
print(
    "This image most likely belongs to {} with a {:.2f} percent confidence."
    .format(class_name[np.argmax(Predict_single_image)], 100 * np.max(Predict_single_image))
)
#print(
#    "This image most likely belongs to {} with a {:.2f} percent confidence."
#    .format(class_name[np.argmax(Predict_single_image_on_valid_data)], 100 * np.max(Predict))
#)
```

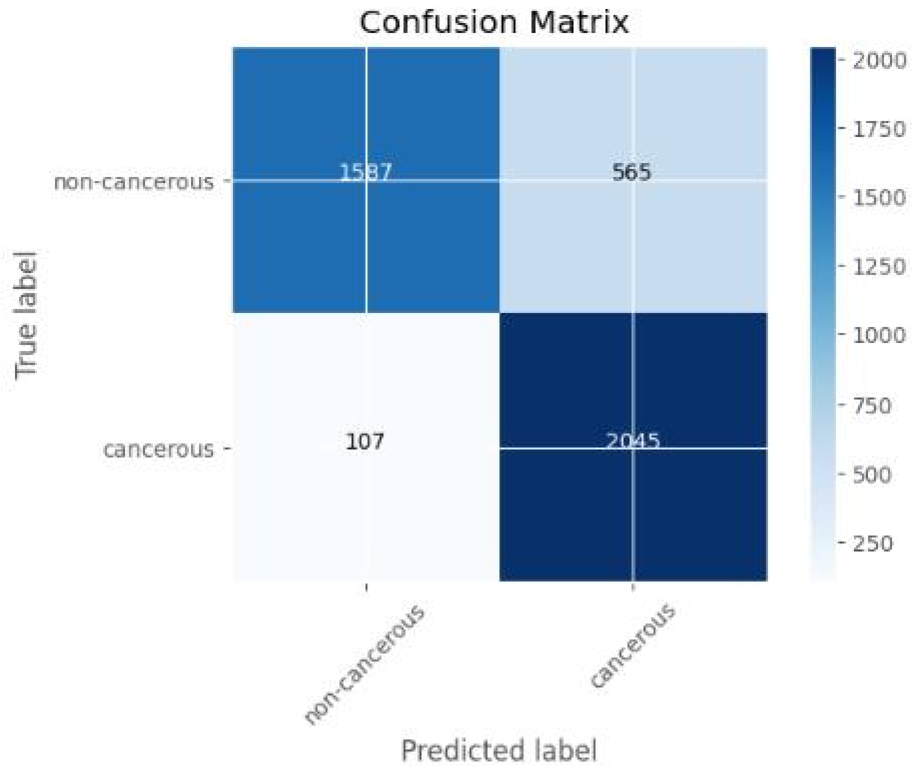
This image most likely belongs to tumorous with a 68.52 percent confidence.

```
for i in predictions:
   print(i)
rounded_predictions= np.argmax(predictions, axis=1)
#rounded_predictions_on_valid_data = np.argmax(predictions_on_valid_data, axis=1)
```

Confusion Matrix for Evaluating the Performance of the Model

```
from sklearn.metrics import confusion_matrix
import itertools
from sklearn import metrics
import sklearn
cm= confusion_matrix(y_true=test_label, y_pred=rounded_predictions)
print(cm)
[[1587 565]
 [ 107 2045]]
def plot_confusion_matrix(cm, classes,
            normalize=False,
            title='Confusion matrix',
            cmap=plt.cm.Blues):
   """
   This function prints and plots the confusion matrix.
   Normalization can be applied by setting `normalize=True`.
   """
   plt.imshow(cm, interpolation='nearest', cmap=cmap)
   plt.title(title)
   plt.colorbar()
   tick_marks = np.arange(len(classes))
```

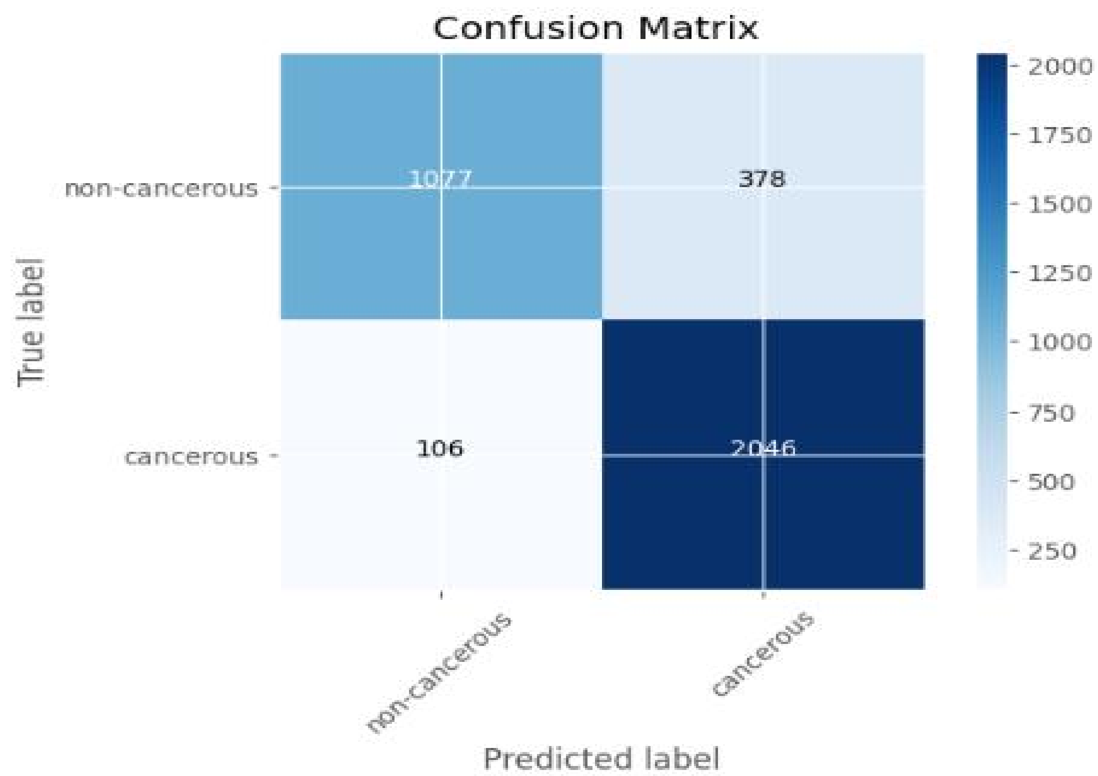
```
    plt.xticks(tick_marks, classes, rotation=45)
    plt.yticks(tick_marks, classes)
    if normalize:
        cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
        print("Normalized confusion matrix")
    else:
        print('Confusion matrix, without normalization')
    print(cm)
    thresh = cm.max() / 2.
    for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
        plt.text(j, i, cm[i, j],
            horizontalalignment="center",
            color="white" if cm[i, j] > thresh else "black")
    plt.tight_layout()
    plt.ylabel('True label')
    plt.xlabel('Predicted label')
cm_plot_labels = ['nontumorous','tumorous']
plot_confusion_matrix(cm=cm, classes=cm_plot_labels, title='Confusion Matrix ')
Confusion matrix, without normalization
[[1587 565]
 [ 107 2045]]
```



Generating Confusion Matrix for Validation Data

```
from sklearn.metrics import confusion_matrix
import itertools
from sklearn import metrics
cm= confusion_matrix(y_true=valid_label, y_pred=rounded_predictions)
print(cm)
[[1059 396]
 [ 112 2040]]
def plot_confusion_matrix(cm, classes,
        normalize=False,
        title='Confusion matrix',
        cmap=plt.cm.Blues):
 """
 This function prints and plots the confusion matrix.
 Normalization can be applied by setting `normalize=True`.
```

```
"                    "                    "
plt.imshow(cm, interpolation='nearest', cmap=cmap)
plt.title(title)
plt.colorbar()
tick_marks = np.arange(len(classes))
plt.xticks(tick_marks, classes, rotation=45)
plt.yticks(tick_marks, classes)
if normalize:
  cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
  print("Normalized confusion matrix")
else:
  print('Confusion matrix, without normalization')
print(cm)
thresh = cm.max() / 2.
for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
  plt.text(j, i, cm[i, j],
    horizontalalignment="center",
    color="white" if cm[i, j] > thresh else "black")
plt.tight_layout()
plt.ylabel('True label')
plt.xlabel('Predicted label')
cm_plot_labels = ['nontumorous','tumorous']
plot_confusion_matrix(cm=cm, classes=cm_plot_labels, title='Confusion Matrix')
Confusion matrix, without normalization
[[1077  378]
 [ 106 2046]]
```



Print Accuracy

```
Accuracy = sklearn.metrics.accuracy_score(y_true=test_label, y_pred=rounded_predictions)
print('Accuracy on test data is: {:5.2f}%'.format(100 * Accuracy) )
Accuracy on test data is: 84.39%
Accuracy_valid = sklearn.metrics.accuracy_score(y_true=valid_label, y_pred=rounded_predictions_valid)
print('Accuracy on validation data is: {:5.2f}%'.format(100 * Accuracy_valid) )
Accuracy on validation data is: 86.58%
import numpy
import sklearn.metrics

r = sklearn.metrics.confusion_matrix(y_true=test_label, y_pred=rounded_predictions)
r = numpy.flip(r)
```

Accuracy = (r[0][0] + r[-1][-1]) / numpy.sum(r)

print('Accuracy is: {:5.2f}%'.format(100 * Accuracy))

Accuracy is: 84.39%

Print Precision

import sklearn.metrics

precision = sklearn.metrics.precision_score(y_true=test_label, y_pred=rounded_predict

ions)

print()

print('Precision on test data is: {:5.2f}%'.format(100 * precision))

Precision on test data is: 78.35%

precision_valid = sklearn.metrics.precision_score(y_true=valid_label, y_pred=rounded

_predictions_valid)

print()

print('Precision on validation data is: {:5.2f}%'.format(100 * precision_valid))

Precision on validation data is: 84.41%

Print Recall

import sklearn.metrics

recall = sklearn.metrics.recall_score(y_true=test_label, y_pred=rounded_predictions)

print('Recall on test is: {:5.2f}%'.format(100 * recall))

Recall on test is: 95.77%

recall_valid = sklearn.metrics.recall_score(y_true=valid_label, y_pred=rounded_predic

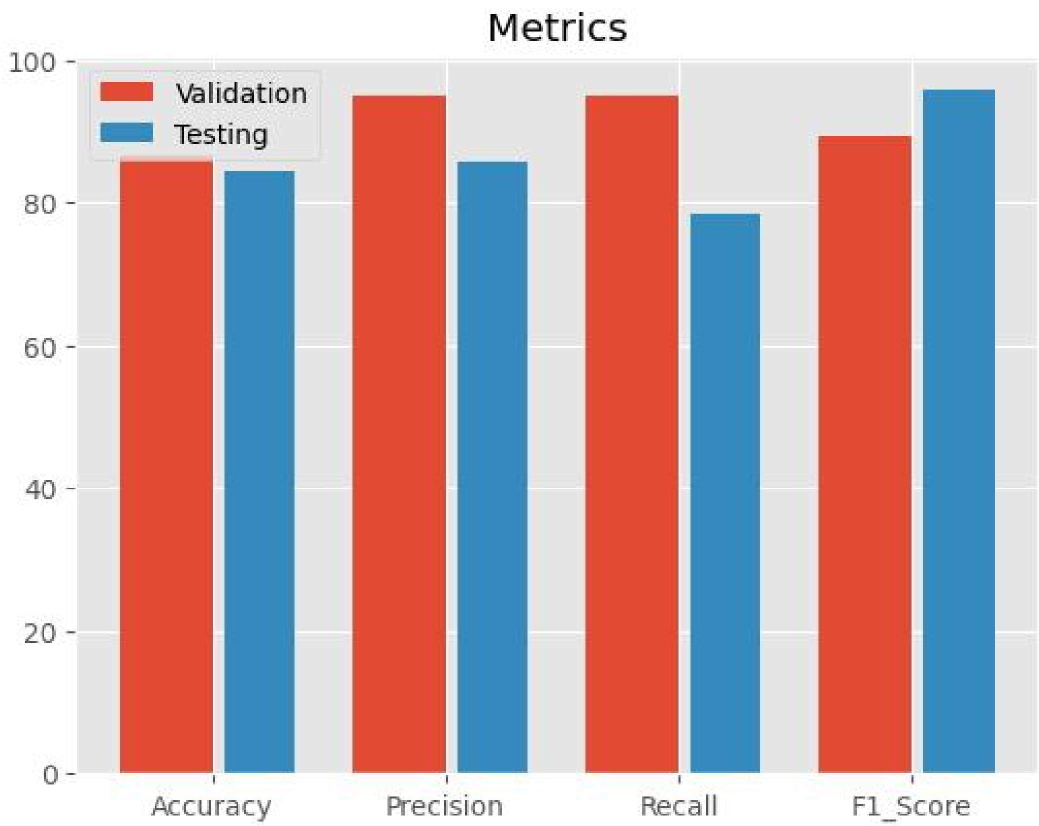
tions_valid)

print('Recall on validation data is: {:5.2f}%'.format(100 * recall_valid))

Recall on validation data is: 95.07%

F1 Score

```
import sklearn.metrics
f1_score = sklearn.metrics.f1_score(y_true=test_label, y_pred=rounded_predictions)
print('Precision is: {:5.2f}%'.format(100 * f1_score) )
Precision is: 85.89%
f1_score_valid = sklearn.metrics.f1_score(y_true=valid_label, y_pred=rounded_predict
ions_valid)
print('Precision on validation data  is: {:5.2f}%'.format(100 * f1_score_valid) )
Precision on validation data is: 89.42%
import numpy as np
import matplotlib.pyplot as plt
X = ['Accuracy','Precision','Recall','F1_Score']
Testing = {84.39 , 78.35,95.77, 85.89}
Validation = [86.58,95.07,95.07,89.42]
X_axis = np.arange(len(X),)
plt.bar(X_axis - 0.2, Validation, 0.4, label = 'Validation')
plt.bar(X_axis + 0.2, Testing, 0.3, label = 'Testing')
plt.xticks(X_axis, X)
#plt.xlabel("Groups")
#plt.ylabel("Number of Students")
plt.title("Metrics")
plt.legend()
plt.show()
```


Metrics
Validation
Testing
100
80
60
40
20
0
Accuracy
Precision
Recall
F1_Score

Bio-data

A. Personal Data

Name:	Kofoworola Folakemi FAMUREWA
Date of Birth:	24-07-1986
Place of Birth:	Ilesa
State of Origin:	Osun State
Nationality:	Nigerian
Address:	Zone 23, Joy Estate, Adegbayi, Ibadan.
Phone Number:	+2348066675105
Email:	ogedengbekofo@gmail.com
Next of Kin:	Ogedengbe Idowu
Address of Next of Kin:	Information Technology Department, University College Hospital, Ibadan.

B. Institutions Attended with Qualifications Obtained and Dates

Lead City University, Ibadan	Ph.D Computer and Information Science 2024
University of Ibadan	Master in Information Science (M. Inf.Sc) 2015
University of Ilorin	B.Sc. Computer Science 2009
Okebode Grammar School	National Examiation Council (NECO) 2004
A.T.C Demonstration	Primary School Leaving Certificate 1996

C. Work Experience with Date

Tower Polytechnic Ibadan	2013 - 2014
Ibadan City Polytechnic	2014 – 2015
School of Health Information Management, University College Hospital, Ibadan.	2015-till date

D. Publications (Published and Unpublished)

a. **Famurewa, K. F. (2009).** Networking of Computers (A case study of faculty of Science, University of Ilorin).

b. **Famurewa, K. F. (2015).** Design of a Mobile Drug Intake Alerting System for Geriatric Patients at the University College Hospital, Ibadan. A Project Submitted for the Award of the Master of Information Science Degree. Africa Regional Centre for Information Science, University of Ibadan, Nigeria.

c. Adeola Omobola Opesade, **Kofoworola Folakemi Famurewa** & Ebelechukwu Gloria Igwe (2017) Gender divergence in academics' representation and research productivity: a Nigerian case study, Journal of Higher Education Policy and Management, 39:3, 341-357, DOI: 10.1080/1360080X.2017.1306907

d. **Famurewa, K. F. (2022):** Assessment of the Initial Adoption and Implementation of Electronic Medical Records (EMR) System in a Nigerian Teaching Hospital. 2022.

e. **Famurewa Kofoworola Folakemi**, Wilson Sakpere, & Adelodun Felicia Ojiyovwi. A Review of Fixed Input Size Limitation in Convolutional Neural Networks Models and Suggested Solutions. 2024.

f. **Famurewa Kofoworola Folakemi**, Wilson Sakpere, & Adelodun Felicia Ojiyovwi. Modified AlexNet Convolutional Neural Network Architecture for Detecting and Classifying Varying Sizes of Brain Tumor MRI Images. 2024.

E. Major Conference with Date

3rd FASCON Conference, 2nd – 4th November, 2022 at the International Conference Centre, Lead City University, Ibadan, Oyo Stste, Nigeria.

F. Referee

Dr. W. Sakpere
Head of Department
Department of Computer Science
Lead City University, Ibadan, Oyo State, Nigeria.
+2348159582869

Signature Date

The University Compliance Certification

This is to certify that this thesis by **Kofoworola Folakemi FAMUREWA** with Matriculation Number **LCU/PG/002338** in the Department of Computer Science, Faculty of Natural and Applied Sciences, Lead City University, Ibadan is in full compliance with the approval of the University's format and style.

……………………………… ……………………

Signature Date