# Performance Evaluation of Deep Learning Models on Brain Tumor MRI Classification and Explainability

**Wandile Judy Nhlapho**
**Student Number: 16020189**

*Supervisor:* Prof Jean-Claude Ndogmo
*Co-supervisor:* Dr Marcellin Atemkeng

submitted in partial fulfilment of the requirements
for the degree of
**Masters of Science in E-Science**

in the

Department of Mathematical and Computational Sciences,
University of Venda, Thohoyandou

Submission Date: July 18, 2024

# Declaration

I, (Wandile Juddy Nhlapho) [student no: 16020189], declare that this research paper is my original work and has not been submitted for any degree at any other university or institution. The research paper does not contain other persons' writing unless specifically acknowledged and referenced as such.

[22/12/2023]

# Abstract

Deep learning models often act as black boxes, making it difficult to understand their decision-making process. To understand how these models make decisions, this paper proposes a framework involving two phases. The first phase evaluates the performance of ten deep transfer learning models—ViT Transformer, EfficientNetB0, DenseNet121, Xception, GoogleNet, Inception V3, VGG16, VGG19, ResNet50, and AlexNet—for classifying brain tumors using MRI scans. The models are assessed based on metrics like accuracy, F1 score, recall, and precision, with EfficientNetB0 outperforming the other models with 98% accuracy and a balanced precision and recall, resulting in an F1 score of 98%. In the second phase, we use interpretability techniques such as Grad-CAM and Grad-CAM++, Integrated Gradient, and Saliency Mapping to investigate what these models learn within MRI images to make classification decisions. The results show that both Grad-Cam and Grad-Cam++ effectively identify the exact locations of tumor localization in the MRI images. This result enhances our understanding of the specific locations within the images where transfer learning models extract features to make classification decisions.

**Key words:** Brain Tumor Classification, Deep learning models, MRI, Explainability, Interpretability, Grad-CAM and Grad-CAM++, Integrated Gradient, Saliency Mapping.

# Acknowledgements

I would like to extend my heartfelt gratitude to God of Mount Sione for granting me the gift of life, unwavering strength, and the wisdom required to complete this academic journey. Your divine guidance has been a constant source of inspiration throughout this study. I am profoundly thankful to my dedicated supervisors, Dr. Marcellin Atemkeng and Professor Jean-Claude Ndogmo, for their invaluable guidance, mentorship, and unwavering support. Their knowledge and dedication to achieving high standards in academics have been crucial in influencing the course and caliber of this study. I am also deeply appreciative of the sponsorship provided by DSI-NEPTTP for my pursuit of a Master's degree in e-science (MSc in e-science). Their kind assistance has allowed me to use valuable resources and seize opportunities that have enriched my scholastic journey. Their motivation and assistance have played a crucial role in my academic development and the successful accomplishment of this study.

# Table of contents

# Dedication

I wholeheartedly dedicate this research study to my beloved son, Wandile Juddy Junior. His presence in my life has consistently served as a wellspring of inspiration and drive, motivating me to not only be a caring father but also a role model who leads through actions. With his firm support and constant reminder of the importance of hard work, he has sparked a fire within me, fueling my unwavering commitment to succeed in both my scholarly and individual endeavors. This firm commitment serves as evidence of the profound influence he has had on my journey, and my aspiration is to continue setting a positive example for him as he grows and matures.

# List of Figures

# List of Tables

# List of abbreviations

**AI** - Artificial Intelligence.
**APR** - Artificial Pattern Recognition.
**BN** - Batch Normalization.
**CNN** - Convolutional Neural Network.
**CT** - Computed Tomography.
**DL** - Deep Learning.
**F1 Score** - Harmonic Precision-Recall Mean.
**FC** - Fully Connected
**GradCam** - Gradient-wieghted Class Activation Mapping
**LIME** - Local Interpretable Model-agnostic Explanations.
**ML** - Machine Learning.
**MRI** - Magnetic Resonance Imaging.
**NPR** - Natural Pattern Recognition.
**ReLU** - Rectified Linear Unit
**RNN** - Recurrent Neural Network
**ROC** - Received Operating Characteristic
**SHAP** - SHapley Additive exPlanation
**SVM** - Support Vector Machine
**TL** - Transfer Learning.
**VGG** - Visual Geometry Group

# Chapter 1

# Introduction

The field of medical imaging has experienced a transformative paradigm shift with the emergence of deep learning (DL), unlocking unprecedented opportunities for accurate disease classification and diagnosis, as discussed in [1, 2, 3, 4, 5]. In the area of brain tumor classification using Magnetic Resonance Imaging (MRI) scan images, these DL models have shown exceptional promise [1, 2]. The significance of brain tumor diagnosis from MRI images lies in its pivotal role in modern healthcare, offering opportunities for early detection and timely treatment, as is evident in [2, 3, 6]. DL models have demonstrated significant promise in automating brain tumor classification tasks. However, their real-world application and reliability, especially concerning accuracy and interpretability, are still questioned [7].

Despite the impressive classification results produced by DL models, their "black-box" nature impedes a comprehensive understanding of the decision-making rationale. Also, their incorporation into clinical practice and decision-making needs adaptability, not just in terms of classification performance, but also in interpretability and explainability. This study investigates the junction of medical image analysis and DL interpretability. Brain tumors, characterized by their diversity and clinical significance as highlighted in [2, 3], demand precise classification for tailored treatment plans and improved patient outcomes. While DL models have exhibited substantial prowess in this domain, their inherent opacity presents a challenge, as discussed in [3, 6]. The central issue revolves around understanding how these models reach their conclusions, particularly crucial in the context of medical decision-making. Today, interpretability techniques are used to understand the decision-making processes of DL models in medical imaging. These techniques aid in identifying crucial features and regions within images that impact the model's decisions, thus enhancing comprehension of the diagnostic process.

We conducted a broad performance evaluation of various state-of-the-art DL frameworks with transfer learning such as AlexNet, DenseNet121, EfficientNetB0, GoogLeNet, Inception V3, ResNet50, VGG16, VGG19, ViT Transformer, and Xception for brain tumor MRI classification. This evaluation encompasses not only the assessment of classification accuracy, but also a focus on evaluating the interpretability of the models. Adaptive path-based techniques, including Grad-CAM++ [8], Grad-CAM [9], and Integrated Gradients [10], are used to understand what these DL models learn in MRI images. Other methods exist in the literature for feature extraction, such as the Stationary Wavelet Transform (SWT) [11] and optimized hybridization methods like Particle Swarm Optimization (PSO) and Artificial Bee Colony

(ABC) [11]. However, in this work, our primary focus is on state-of-the-art CNNs using a gradient-based optimization method.

## 1.1 Scientific contribution of the study

This study of "Performance Evaluation of DL Models on Brain Tumor MRI Classification and Explainability" makes significant scientific contributions in several areas such as:

- *Model Evaluation*: The study comprehensively assesses various DL architectures, offering valuable insights into which models are most effective for brain tumor classification. This evaluation is key for guiding the selection of appropriate models in real-world medical applications.

- *Brain Tumor Diagnosis:* Diagnosing a brain tumor is a challenging process that need the correct and rapid examination of MRI scan images. The study's findings directly contribute to enhancing the accuracy and reliability of DL models for identifying brain cancers by focusing on this specific medical area. This is critical for early diagnosis and treatment planning for patients.

- *Model Interpretability:* The incorporation of explainability approaches, like Grad-CAM, Grad-CAM++, Integrated Gradient, and Saliency Mapping, represents an important scientific contribution. By using these methods, the study increases the interpretability of DL models, clear up on the decision-making processes and gives valuable intuition into how these models arrive at their classifications, especially in the context of brain tumor diagnosis.

## 1.2 Research questions

• How do different DL models perform in classifying brain tumors from MRI scan images, and how does their performance compare in terms of accuracy, precision, recall and F1-score?

• To what extent do advanced interpretability methods, such as Grad-CAM, Grad-CAM++, Integrated Gradients and Saliency Mapping, enhance the transparency of DL model's decisions in brain tumor classification?

## 1.3 Research aim and objectives

### 1.3.1 Aim

The aim of the study is to evaluate various DL architectures for the classification of brain tumors based on MRI scan images and to investigate the interpretability of these models using Grad-CAM, Grad-CAM++, Integrated Gradient and Saliency Mapping techniques.

### 1.3.2 Objectives

• To assess the performance of diverse DL architectures for the accurate classification of brain tumors using MRI scan images.

- To explore the interpretability of these DL models through the application of Grad-CAM, Grad-CAM++, Integrated Gradient and Saliency Mapping techniques, aiming to understand their decision-making processes.

## 1.4 Thesis Organization

Chapter 2, titled "Machine Learning and Deep Learning". It commences with Section 2.1, "Introduction to Machine Learning," which defines the fundamental concepts of ML. Our exploration covered a detailed examination of supervised and unsupervised learning, Additionally, we dig in into the principles and significance of two influential classification techniques, Support Vector Machines (SVM) and Random Forest. The chapter then delves into a detailed exploration of Section 2.2, "Introduction to Deep Learning," We began by introducing key concepts in DL, with a specific focus on DL as the central theme of our study. DL, acknowledge for its self governing feature learning competences, was introduced, laying the groundwork for our subsequent investigation into specific DL models, such as AlexNet, ResNet50, Xception, DenseNet121, GoogleNet, Inception V3, VGG16, VGG19, Vision Transformer ViT and EfficientNetB0. To ensure the robustness of our model assessments, we introduced essential transfer learning (TL) method and performance evaluation techniques.

Chapter 3, titled "Literature on Explainability". It commences with Section 3.1, "Introduction to Explainability and Interpretability," which defines the fundamental concepts and their significance. The chapter then delves into a detailed exploration of Section 3.2, "Techniques for Explainability and Interpretability," focusing on methods such as Grad-CAM, Grad-CAM++, Saliency maps and Integrated Gradients (IG). Section 3.3, "Interpretability in Medical Imaging," follows, emphasizing practical applications and real-world case studies. The subsequent section, Section 3.4, "Quantifying Interpretability Metrics," discusses methodologies for measuring interpretability. Section 3.5, "Integrating Interpretability in Clinical Practice," scrutinizes the integration of interpretable models into healthcare workflows, while Section 3.6, "Adaptive Path-Based Techniques and Challenges," explores advanced methods. The chapter concludes with Section 3.7, "Future Directions," offering insights into emerging trends and unresolved challenges in the field, effectively framing the subsequent empirical research chapters of the thesis.

Chapter 4, titled "Materials and Methodology," is structured into two essential sections. Section 4.1, "Dataset of the Study and Study Approach," provides a thorough exploration of the dataset used, encompassing its origin, characteristics, and preprocessing steps, while also elucidating the study approach, encompassing the methodology for data collection and ethical considerations. This section establishes the groundwork for the study's context and scope. In Section 4.2, "Experimental Setup," the technical facets of the research design are revealed, including hyperparameters, training configurations, and software utilized in the experiments. These sections collectively offer a transparent and comprehensive insight into the research methodology, from data acquisition to the experimental framework, ensuring both clarity in the research process and the potential for reproducibility.

Chapter5, titled "Results and Discussions," is composed of pivotal sections that reveal the study's findings and engage in extensive discussions. In Section 5.1 "Classification results" we begin by discussing the training results, and offering insights into the model training process. Then focus shifts to classification outcomes and their implications, providing insights into the model's efficacy in diagnosing brain tumors. Section 5.2, "Interpretability results and Discussion" presents the insights obtained through interpretability techniques, shedding light on the decision-making processes of the models and their practical implications. Finally, the section titled "Conclusion" distills the key findings, offering a condensed overview of the study's results and their broader significance.

# Chapter 2

# Machine and Deep Learning Methods

Chapter 2 serves as a comprehensive foundational chapter in our study. It commences with an outline of ML in Section 2.1, which includes detailed discussions on supervised and unsupervised learning. In Section 2.2, the focus shifts to DL, where neural networks and their architectural variations are explored. Section 2.2.3 highlight the importance of transfer learning as a means to make sure the robustness and reliability of our models. In Section 2.2.4, the central theme of our study, performance evaluation, is addressed comprehensively. This section scoops out into a wide range of evaluation metrics and methodologies essential for assessing the effectiveness and efficiency of DL models. Finally, Chapter 2 concludes with Section 2.3, summarizing the key points covered and setting the stage for the subsequent chapters in our study.

## 2.1 Introduction to Machine Learning

Before we embark on our exploration of DL, it is crucial to establish a foundation in the broader domain of ML. A revolutionary area of AI is ML, which enables enables the magnificent machines to acquire knowledge from vast amounts of data and astutely generate forecasts or discernments without the need for explicit instructions [7, 8]. Applications for this vary greatly, encompassing a broad spectrum, which includes image recognition and recommendation systems to medical diagnosis and self-driving cars [7, 9]. The generality idea of ML is to bestow computers the ability to identify patterns and to get better over time as more data becomes available. ML differs from conventional rule-based programming in that it can learn and adapt [8].

### 2.1.1 Supervised Learning

The ML concept of supervised learning is crucial [9]. In supervised learning, a model is developed using a labeled dataset, where each input data point has a matching target or result. By reducing the gap between its predictions and the actual targets, the model can learn to map inputs to outputs [9, 10]. By doing this, the model can apply its expertise more broadly and generate predictions about brand-new, untested data. In missions where the purpose is to predict a continuous or categorical outcome, supervised learning is frequently applied. Exemplification of these missions include image classification, text sentiment analysis, and regression problems. To illustrate this concept, let's consider two common examples. First, in email spam detection, the objective is to classify emails as either spam or not spam (ham). Using a dataset of labeled emails, where each email is marked as spam or ham, the algorithm learns patterns and features indicative of spam emails during training. These features might include keywords, sender information, or email format. Once trained, the model can accurately classify new, unseen emails as spam or ham based on their features. Second, in house price prediction, the goal is to estimate the price of a house based on its attributes such as size, number of bedrooms, and location. Using a dataset of labeled house listings containing both the features and sale prices, the algorithm learns the relationship between the features and the sale price during training. This allows the model to predict the price of new, unseen houses accurately.

#### 2.1.1.1 Classification Algorithms

In image classification make use of supervised learning, Support Vector Machines (SVMs) and Random Forests are two memorable algorithms known for their influence in various scenarios. Support Vector Machines (SVM) is engaged for both classification and regression applications. In the context of classifying images, SVM creates hyperplanes that best divide several classes in feature space. By maximizing the distance between classes, these hyperplanes enhance the art of generalization and the essence of robustness. The widest margin (distance) between the support vectors of various classes the data points that are closest to the decision boundary is what the SVM looks for in a hyperplane, as shown in Figure 2.1. Finding the hyperplane that minimizes the chance of incorrect categorization on unknown data while maintaining a safe margin is the goal [10].

SVM's capacity to manage intricate decision boundaries is one of its main

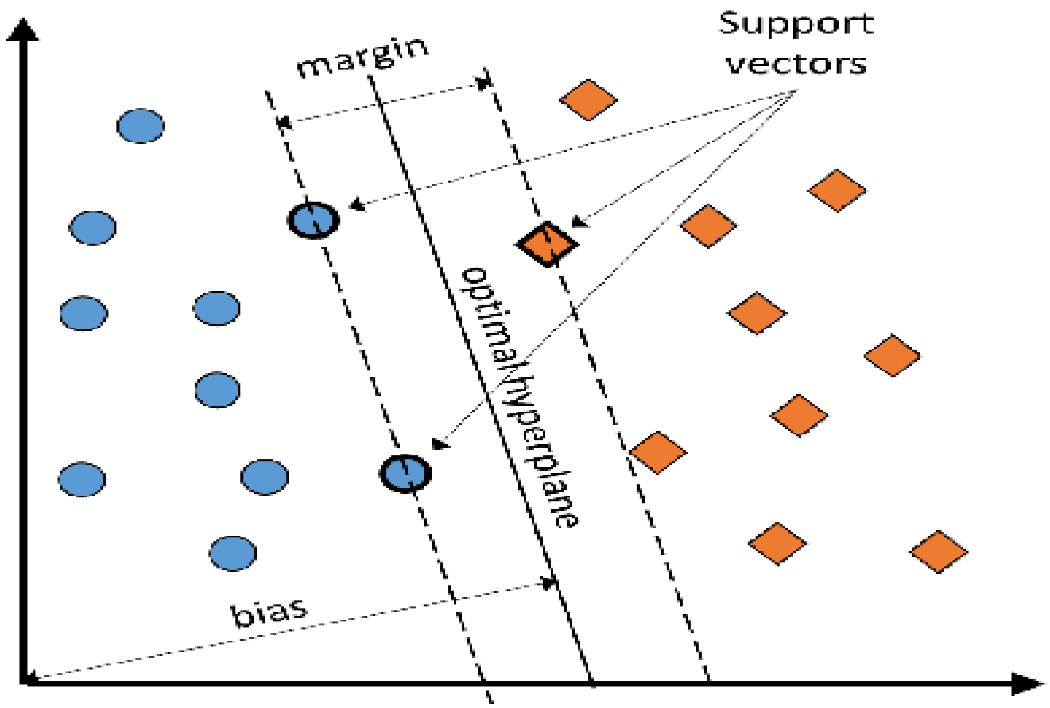


Figure 2.1: Optimal Separating Hyperplane in SVM Adapted from [10].

benefits. When data cannot be separated linearly in the original feature space, SVM is especially useful. SVM can map data into a greater-extent space where it is separable by utilizing kernel functions (for example, polynomials, radial basis functions) [10, 11]. SVM is specially suitable for medical image classification problems because of this capabilities, where differentiating complex characteristics may call for non-linear decision limits. Yet SVM has lot of advantages, but there are short comings as well. Its sensitivity to noisy data and outliers is a noteworthy drawback. Outliers close to the decision boundary have a major impact on where the hyperplane is located, potentially producing less-than-ideal outcomes. Additionally, SVM is less effective for large datasets because its computational complexity rises with dataset growth [12]. SVM is also a very high variance estimator.

Random Forest is a method of a group learning that combines the predictions of numerous decision trees [13]. When dealing with intricate, greater-extent data, it is extremely versatile and effective in picture/image categorization. Random Forest reduces overfitting by combining the outputs of many decision trees, each trained on a distinct sample of the data [13, 14]. This group learning technique improves the model's robustness and generalizability. Random Forests are ideal for picture/image classification applications that require huge datasets with a variety of complex features [13]. They have found use in tasks such as natural scene object recognition, where the fluctuation in visual content and context necessitates a resilient categorization technique.

Random Forest is a highly accurate and robust ensemble learning system that makes predictions by combining numerous decision trees[13, 14]. Its strengths are its power to handle complex datasets with features and missing values, as well as its defiance to overfitting. It gives useful metrics for determining the relevance of features for feature selection [14]. However, the black-box aspect of Random Forest and its computing run on can make it less interpretable and resource-intensive, particularly when working with a large number of trees. Furthermore, it may not be the best solution for datasets with mostly linear relationships, when simpler models may outperform. To build upon our understanding of machine learning paradigms, let's now delve into unsupervised learning, a complementary approach that focuses on uncovering patterns and structures within data without the need for explicit labels.

### 2.1.2 Unsupervised Learning

Unsupervised learning is a core machine learning paradigm that concentrates on discovering patterns, structure, or information within data without direct supervision or labeled results [13, 14]. When handling huge, unannotated datasets, this method is quite helpful. We will give a thorough introduction to unsupervised learning, including its methods and uses, below.

Unsupervised learning is a method for machine learning in which the model gains knowledge from unlabeled data by spotting patterns or relationships [15]. There are no predetermined target labels, in contrast to supervised learning. Instead, the model seeks to elucidate latent representations or structures in the data. For instance, in customer segmentation, businesses utilize unsupervised learning techniques such as clustering algorithms to group customers with similar characteristics or behaviors together. By analyzing customer data without predefined labels, companies can identify distinct customer segments like frequent buyers, occasional shoppers, and bargain hunters. This enables targeted marketing strategies and personalized product recommendations, enhancing customer satisfaction and retention.

Another example of unsupervised learning is anomaly detection in network security. Network administrators leverage techniques like autoencoders and isolation forests to analyze network traffic data without labeled anomalies. By monitoring normal patterns of network behavior, unsupervised learning algorithms can detect deviations such as sudden spikes in traffic or unusual communication patterns, flagging them as potential security threats for further investigation and mitigation. Through unsupervised learning, organizations can bolster their cybersecurity measures and safeguard sensitive information from potential attacks. Having established a foundation in machine learning principles and techniques, let's now delve into the advanced domain of deep learning, which extends the capabilities of traditional machine learning models by harnessing complex neural networks to tackle more intricate tasks and extract higher-level features from data.

## 2.2 Introduction to Deep Learning

Now that we are entering the core of our research, it is crucial to investigate the fascinating world of DL. Our method for resolving complicated problems has been completely changed by DL, a branch of ML [15, 16, 17, 18]. Artificial neural networks (ANN), modeled after the structure and operation of the human brain, form its core. A simplified example of an ANN is presented in Figure 2.3. DL is unique in that it can learn hierarchical features and representations automatically from unprocessed data, doing away with the requirement for labor-intensive manual feature engineering [19]. DL models, also known as neural networks (NN), are particularly well-suited for image and audio recognition, natural language processing, and autonomous decision-making because of their intrinsic versatility.

### 2.2.1 Neural Networks

ANNs, usually referred to as neural networks, are a type of ML models that were influenced by the composition and operation of the human brain [10]. They are made up of interconnected nodes, which are arranged into layers and are also referred to as artificial neurons or perceptrons. The popularity of neural networks has grown as a result of their capacity to extract intricate patterns and representations from data [8, 20]. We'll provide a detailed overview of NN, their components, and their functioning.

The fundamental processing element of a neural network is the neuron, as illustrated in Figure 2.2. Every neuron receives one or more inputs, processes those inputs, and then outputs the result of that calculation. The biological neurons seen in the human brain are a rough representation for how neurons work [21]. A NN has layers that are made up of neurons. Typically, there are three sorts of layers: The neurons in input layer do little more than receive inputs and pass them on to the following layer. The input layer should have the same number of layers as the attributes or features in the dataset [21, 22]. Depending on the type of model, there are hidden layers between the input and output layers. Large numbers of neurons are present in hidden layers. Before sending the inputs to the following layer, the neurons in the hidden layer alter them. The network's weights are modified as it is trained to become more predictive. Via a system of weighted connections in the hidden layer, the data is really processed. The output layer is connected to the hidden layers. In a classification problem, the output layer is the anticipated feature, or class, and it mostly depends on the type of model that was created. Based on the data passed from the hidden layer, the output layer generates the output.

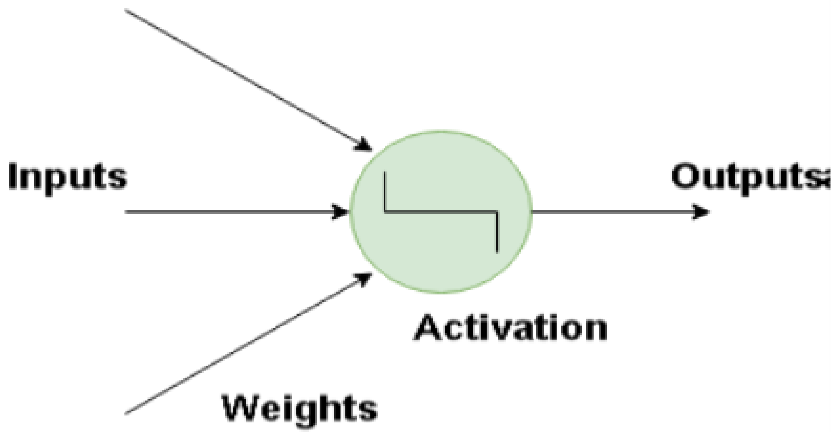


Figure 2.2: An Image of a Neuron Adapted from [21].

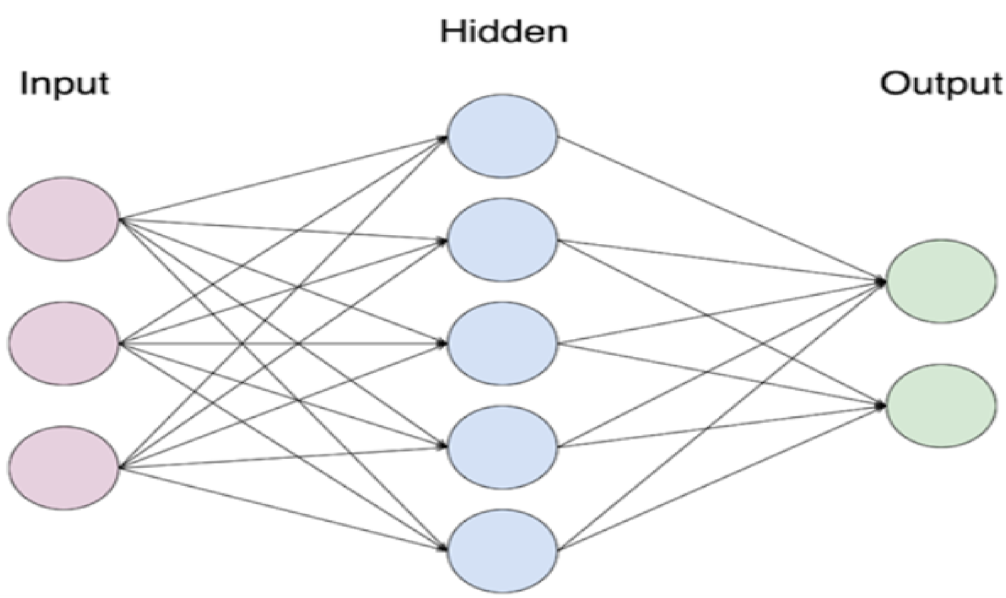


Figure 2.3: A Simple Artificial Neural Network (ANN) with one hidden layer, adapted from [21].

In the 1980s, Kunihiko Fukushima established the fundamental concept of the convolutional neural network CNN [21, 22]. CNN are a class of NN that have demonstrated outstanding performance in domains like image recognition and classification. CNN are the dominant technology in computer vision because they are accurate at classifying images. CNNs are a subset of deep, feed-forward artificial neural networks that use a multi-layer perceptron variant with a low pre-processing requirement (connections between nodes do not form a cycle) [16, 23]. Since CNN architectures explicitly assume that the inputs are images, we can embed specific features into the architecture. These therefore greatly reduce the number of parameters in the network and improve the efficiency with which the forward function can be implemented. Neurons that make up CNN have biases and weights that can be learned. Each neuron processes a few inputs, conducts a dot product, and may optionally do a non-linearity as a follow-up. From the unprocessed image pixels at one end to the class scores at the other, the entire network continues to represent a single differentiable score function. And the final layer has a loss function [24].

The foundation of a CNN is the convolutional layer, and Figure 2.4 gives a visual representation of a CNN structure designed for image classification. It has some distinguishing qualities. The majority of the computing work is handled by it. A group of learnable filters make up the convolutional layer's parameters. Each filter, although its small width and height, thoroughly covers the depth of the input volume. Another element of CNN is the pooling layer. Pooling layers are typically added after convolutional layers. Its pur-

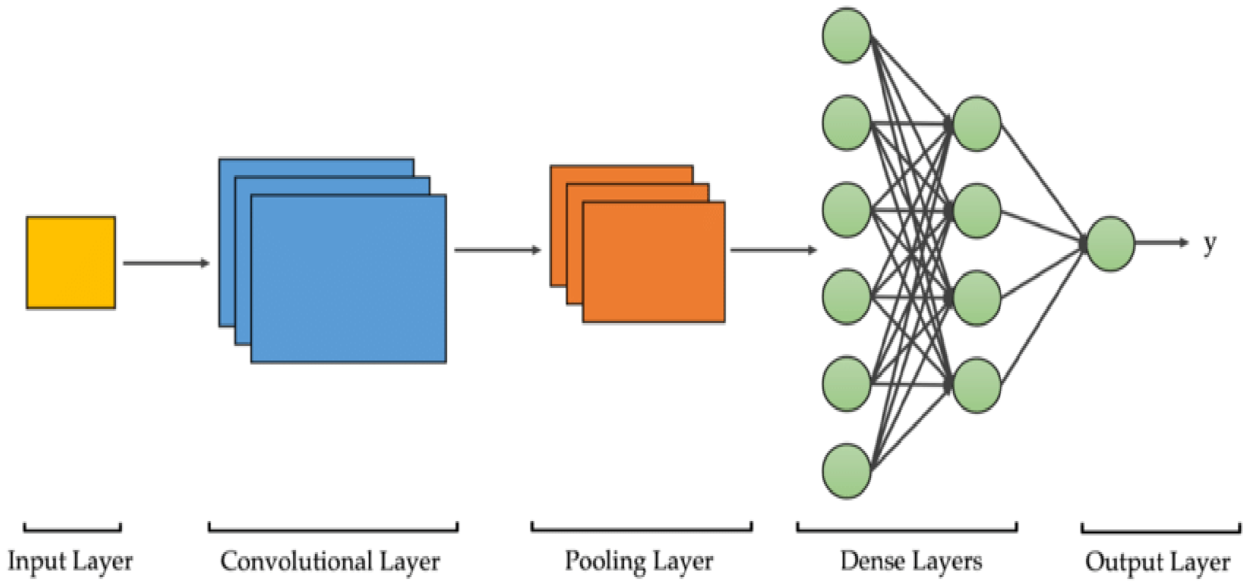


Figure 2.4: A fundamental convolutional neural network (CNN) structure for classifying images adapted from [24].

pose is to gradually shrink the representation's spatial size in order to lessen the number of parameters and computations required in the network and, as a result, to control overfitting. Every depth slice of the input is independently processed by the Pooling Layer, which also resizes the input spatially. The depth dimension of the input volume is unaffected by pooling. To add non-linearity to neural networks, activation functions are used. It constricts the values to a narrower range, like sigmoid activation function compresses numbers in the 0 to 1 range [24, 25]. In order to use larger learning rates, batch normalization (BN) is a technique to lessen the internal covariate shift in neural networks. The approach, in theory, adds a step after each layer in which the output from the layer before is normalized. The network becomes even faster thanks to BN's additional prevention of smaller parameter changes from amplifying.

### 2.2.2 Types of Architectures Used in the Study

In this section, we will look into the DL model's architectures that are crucial to our research. We will look at AlexNet, ResNet50, DenseNet121, GoogleNet, Xception, Inception V3, VGG16, VGG19, Vision Transformer and EfficientNetB0.

#### 2.2.2.1 AlexNet

Deep CNN architecture AlexNet made significant contributions to the development of DL and computer vision. By winning the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2012, it was created by Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton and considerably advanced the state-of-the-art in picture/image classification problems [26]. The success of AlexNet paved the path for more intricate convolutional neural network architectures and served as the catalyst for numerous other developments in DL for computer vision. Modern neural network models are still influenced by its design tenets, such as the usage of convolutional layers, ReLU activations, and dropout regularization.

#### 2.2.2.2 ResNet50

The ResNet50 design combines a combination of convolution filters of various sizes to address the degeneration of CNN models and shorten training times [27]. A maxpool layer, an average pool layer, and a total of 48 convolutional layers make up this architecture.

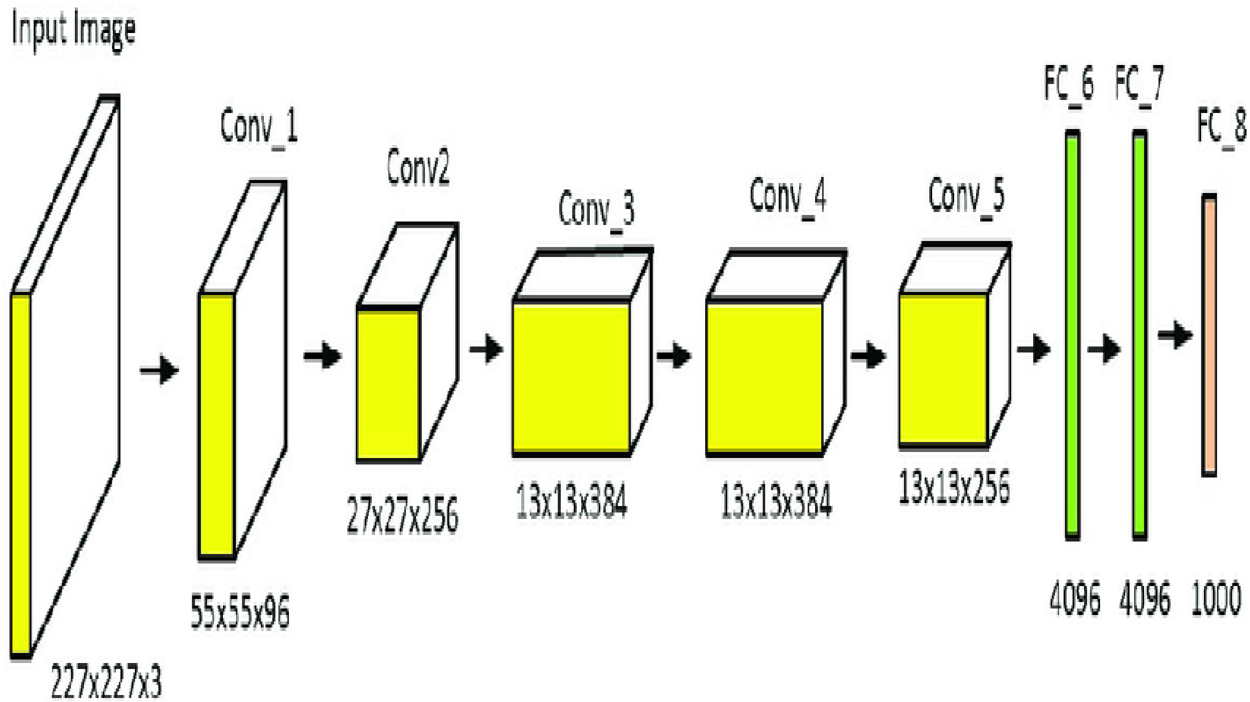


Figure 2.5: Architecture of Alexnet Adapted from [27]. It consisting of eight layers, including five convolutional layers with different filters followed by max-pooling, and three fully connected layers.

#### 2.2.2.3 DenseNet121

The DenseNet model was proposed by [13], whose primary components are DenseBlock (DB), transition layer, and growth rate. A series of devices under the DenseNet brand include the DenseNet121, DenseNet169, DenseNet201, and DenseNet264. DenseNet121 has the advantage of requiring fewer parameters, allowing for the training of deeper models during computation. Additionally, the fully connected layer of the model also has a regularization effect, which can help prevent overfitting on smaller datasets. The network architecture of DenseNet121, which has 427 layers and 7,040,579 parameters.

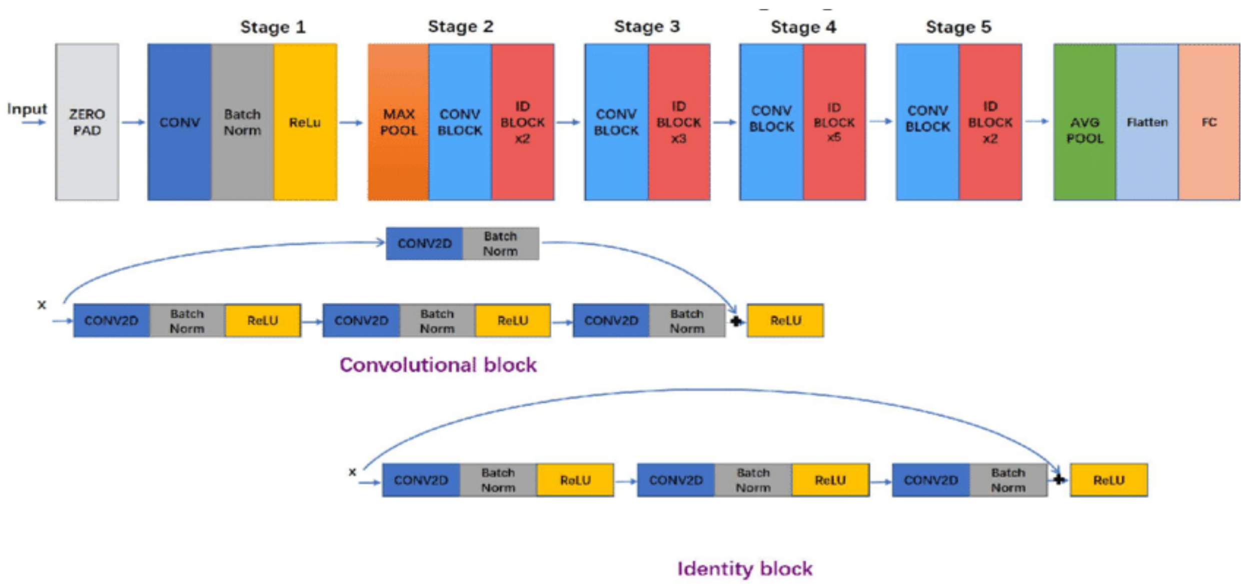


Figure 2.6: Architecture of ResNet50 Adapted from [27]

#### 2.2.2.4 Xception

Inception V3 was updated by Google to create Xception [12], which split regular convolution into spatial convolution and point-by-point convolution. Depthwise Separable Convolution (DSC) was used in place of the original Inception module. While pointwise convolution utilizes a 1x1 kernel to convolve point by point, reducing the number of parameters and computations, spatial convolution is conducted on each input channel. The 132-layer, 20,867,627-parameter Xception network architecture used in this study is made up of 14 modules, 36 of which are convolutional layers. The input image size for this model is 299x299 pixels.

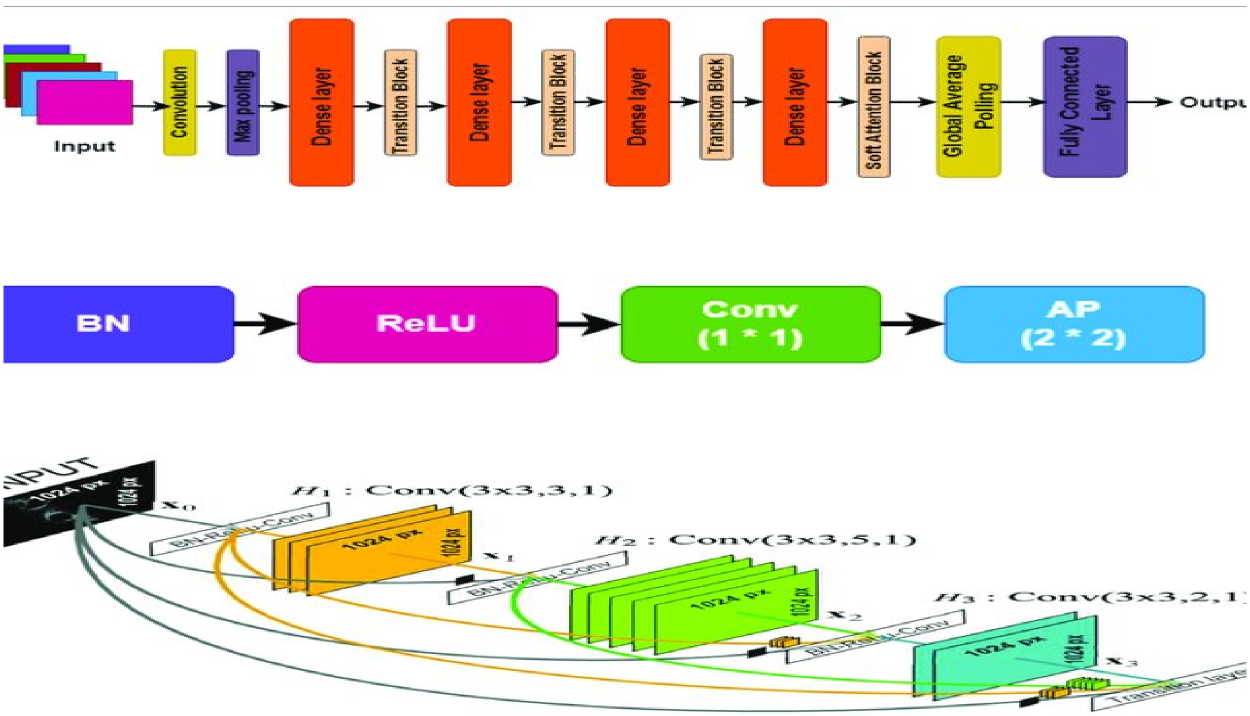


Figure 2.7: Architecture of DenseNet121 Adapted from [18]

#### 2.2.2.5 Inception V3

Inception V1 [19] was the name given to Google's initial proposal of GoogLeNets, which was followed by Inception V2 and Inception V3 [23, 24] the following year. To reduce the size of feature maps, Inception V3 employs convolutional layers with a stride of 2 in combination with pooling layers. The first Inception module of Inception V3 replaces the 7x7 layer convolutional layer with 3by3 layer convolutional layers, which is a modification from Inception V2's first Inception module. The network's width and depth are increased in Inception V3 with the aforementioned upgrades to enhance performance. Additionally, the image's dimensions were changed from the original Incep-

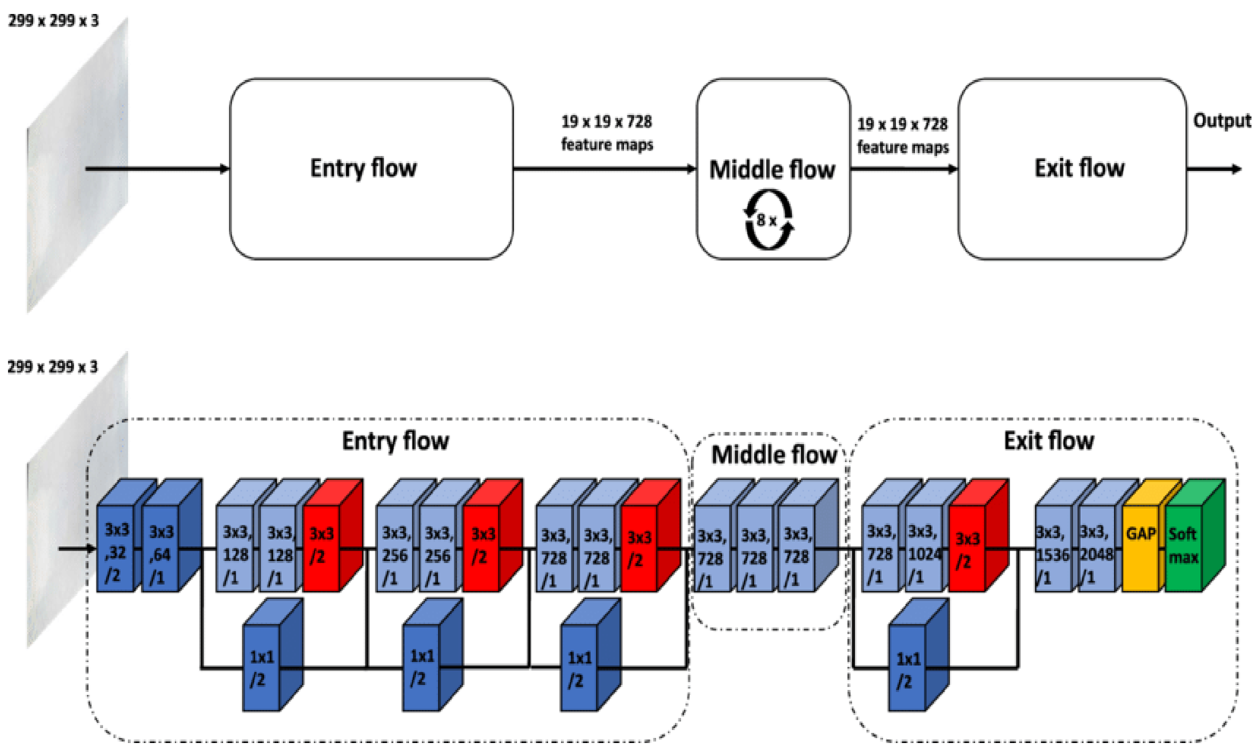


Figure 2.8: Architecture of Xception Adapted from [18]

tion V1's 224x224 to 299x299 pixels. A network design with 311 layers and 21,808,931 parameters.

#### 2.2.2.6 VGG16

Modern transfer learning model VGG16 has sixteen weighted layers and is state-of-the-art. For the top five test results in the ImageNet dataset, the model had an accuracy rate of 92.7%. Additionally, it took first place in the Large-Scale Visual Recognition Challenge (ILSVRC) contest, which was held by the Oxford Visual Geometry Group. The VGG model's greater depth allows the model to assist the kernel in learning more intricate features.

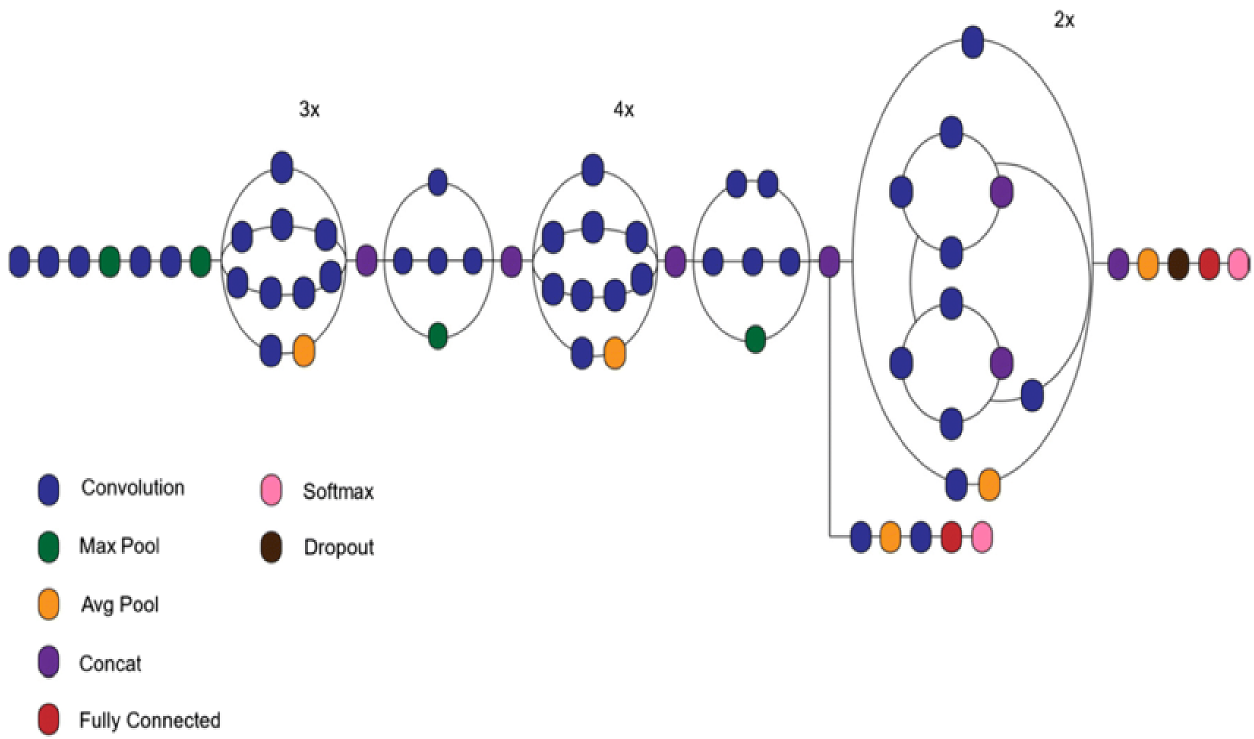


Figure 2.9: Architecture of Inception V3 Adapted from [21]

#### 2.2.2.7 VGG19

In the VGG19 model, which has 19 weighted layers in addition to the VGG16 model plus three extra FC layers with a combined total of 4096, 4096, and 1000 neurons, there are 19 weighted layers in all. A Softmax classification layer and five Maxpool layers are also present. The convolutional layers employ the ReLU activation function.

#### 2.2.2.8 GoogleNet

The best-performing model, GoogLeNet, was revealed by Google at the 2014 ImageNet Large-Scale Visual Recognition Challenge (ILSVRC14) [20]. The

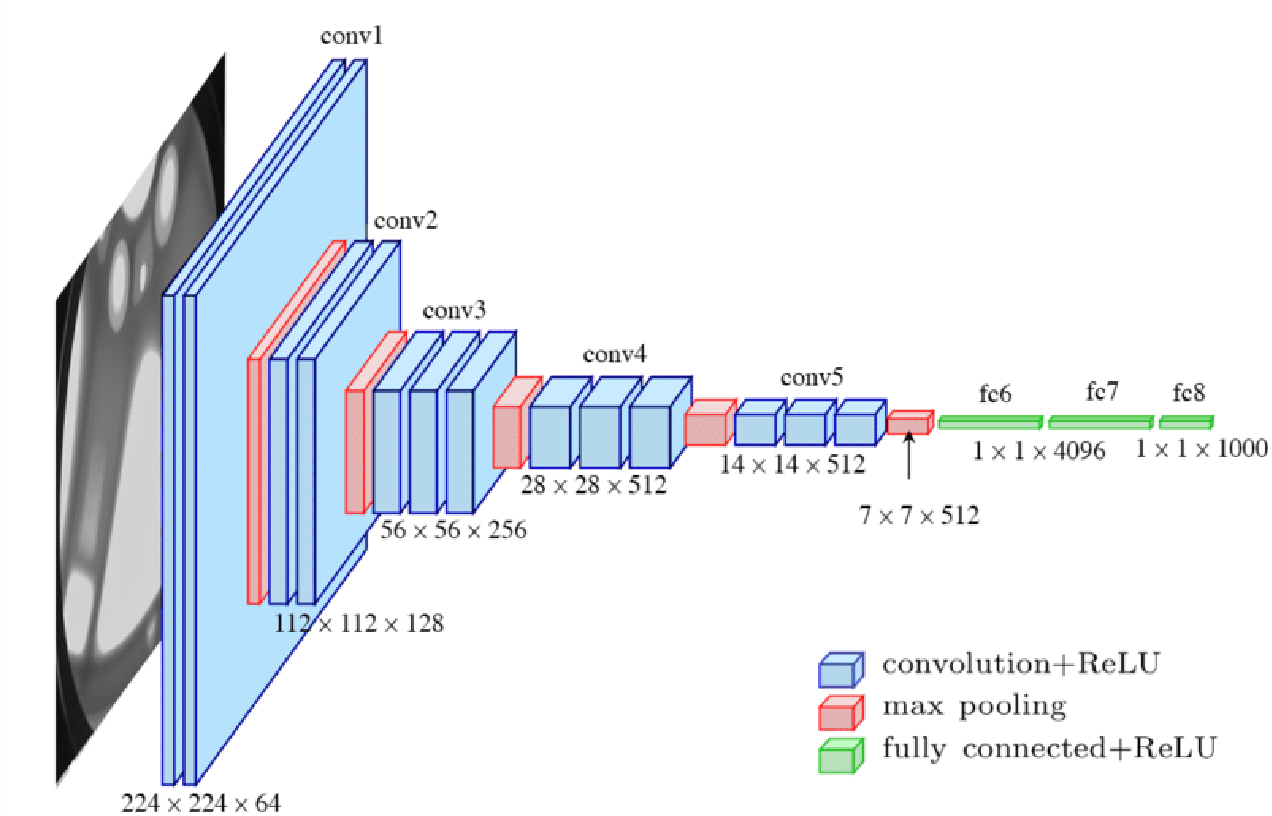


Figure 2.10: Architecture of Vgg16 Adapted from [22]

inner layer of the neural network was expanded to output different correlation distributions at the heart of this structure on the theory that if different probability distributions with high correlations to the input data are attained, the neural network output of each layer will have the best efficiency.

#### 2.2.2.9 Vision Transformer ViT

The architecture adopts the transformer's encoder component, revolutionizing image processing by segmenting the image into patches of a specified

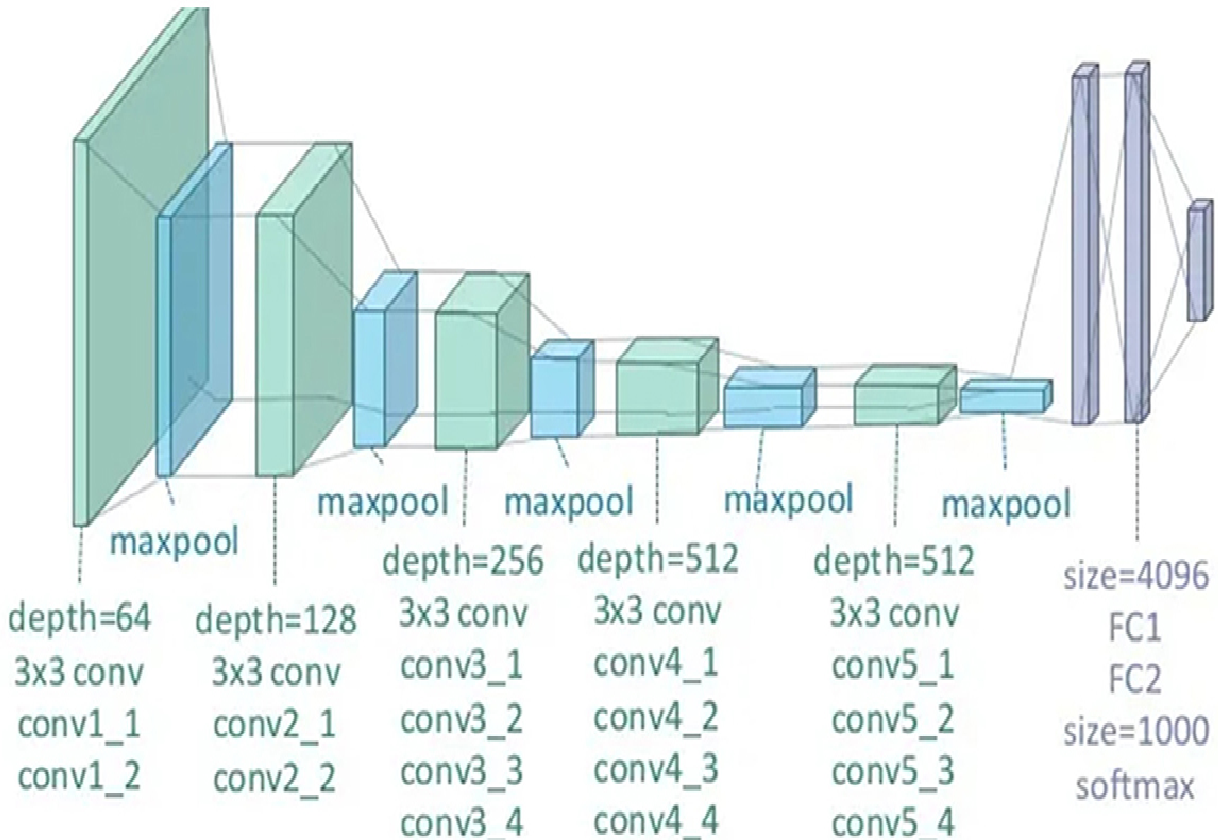


Figure 2.11: Architecture of Vgg19 Adapted from [21]

size like 16x16 or 32x32 dimensions [28]. This patch-based method enhances training with smaller patches. After flattening, patches are fed into the network. Unlike traditional neural networks, the model lacks positional information about the sequence of samples. To address this, the encoder incorporates trainable positional fixed vectors, eliminating the need for hard-coded positions [25].

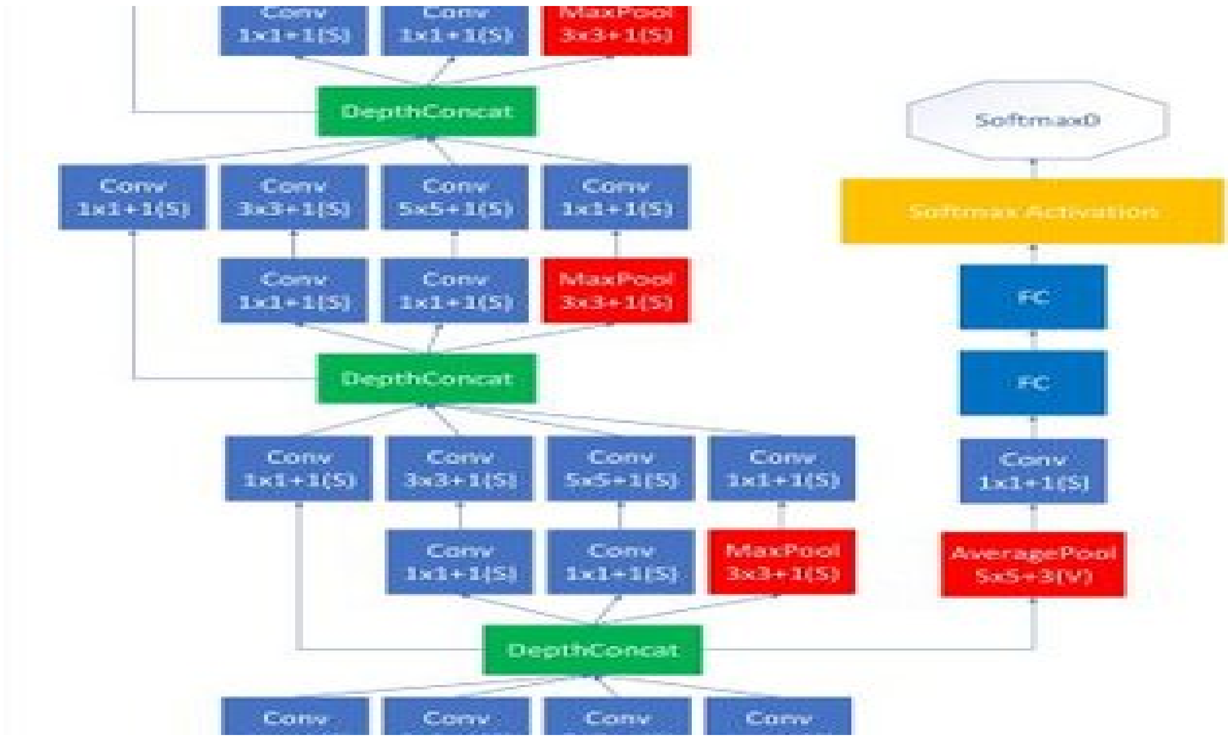


Figure 2.12: Architecture of GoogleNet Adapted from [19]

#### 2.2.2.10 EfficientNetB0

The EfficientNetB0 CNN architecture was proposed in [29]. Enhancing accuracy and efficiency through balanced scaling of the model's depth, breadth, and resolution is the aim of EfficientNetB0. The design presents a fixed ratio compound scaling technique that scales all three dimensions of depth, breadth, and resolution consistently.

### 2.2.3 Transfer Learning (TL)

The concept of TL can be likened to the notion of drawing upon previous experiences and applying them to novel, interconnected assignments. Consider

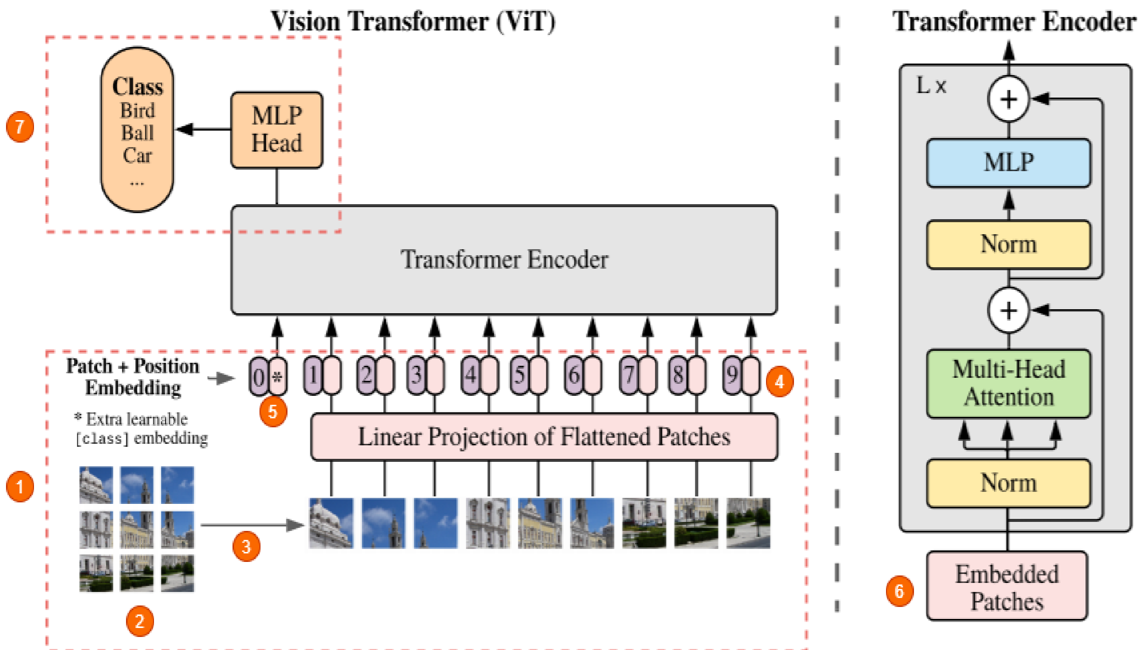


Figure 2.13: Vision Transformer Architecture Adapted from [25]

a situation in which one have expertise in playing a certain musical instrument, such as the flute, and now wishes to agglomerate aptitude in another instrument, such as the xylophone. By virtue of one's prior encounter with the flute, invaluable knowledge pertaining to music theory, the interpretation of sheet music, and the comprehension of melodies has been acquired. In the realm of artificial intelligence, TL bears similarities to this analogy. It entails the utilization of a pre-existing AI model, which has been trained to identify fishes in images, for instance, as a foundation for a distinct yet related task, namely the identification of fogs in images. This approach serves to economize time and resources, much in the same manner as the avoidance of having to relearn music theory from scratch when delving into a new instrument.

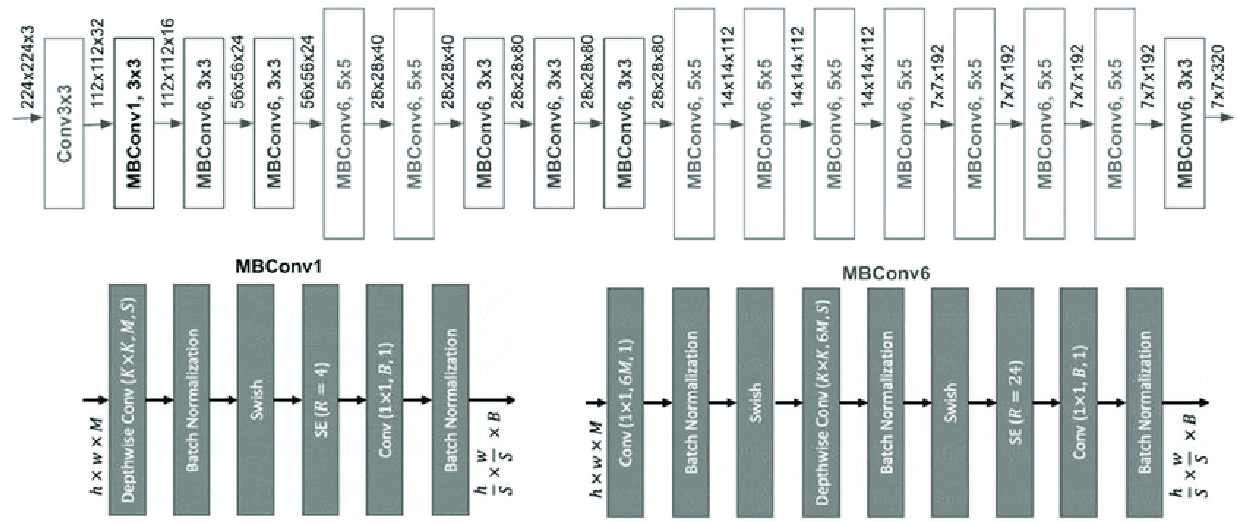


Figure 2.14: EfficientNetB0 Architecture Adapted from [29]

When faced with having a model excel in a related, yet slightly different challenge, transfer learning (TL) becomes pertinent. Rather than training the model from the initial stage, this approach involves leveraging the acquired knowledge, specifically the parameters, from the initial task and applying it to the new, related task. This process facilitates a more efficient adaptation of the model to the novel task by capitalizing on the previously acquired expertise. The TL models under consideration include AlexNet, VGG16, VGG19, GoogLeNet, ResNet50, Inception V3, DenseNet121, Xception, EfficientNetB0 and Vision Transformer (ViT).

### 2.2.4 Performance Evaluation

Some of the most important performance measures frequently used are accuracy, precision, recall, and the so-called $F_1$ score. To describe these measures, let us denote by $\mathrm{TP}, \mathrm{TN}, \mathrm{FP}$, and FN the number of true positives, true negatives, false positives, and false negatives, respectively. Here, TPs are instances where the model accurately predicted the presence of the positive class. TNs are instances where the model accurately predicted the absence of the negative class. FPs are instances where the model predicted a positive outcome when it ought to have been negative. FNs are instances where the model predicted a negative outcome when it ought to have been positive. The performance measures are defined as:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{2.1}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2.2}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2.3}$$

$$F_1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{2.4}$$

Suppose we have developed a machine learning model to classify emails as either spam or not spam (ham). After training and testing the model, we obtain the following confusion matrix:

| | Predicted Spam | Predicted Ham |
|---|---|---|
| Actual Spam | 85 | 15 |
| Actual Ham | 10 | 190 |

From this confusion matrix, we can calculate precision and recall as follows:

**Precision**: Precision measures the proportion of correctly predicted spam emails among all emails predicted as spam.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

In our example:

$$\text{Precision} = \frac{85}{85+10} = \frac{85}{95} \approx 0.8947$$

So, the precision of our model for predicting spam emails is approximately 0.8947 or 89.47%.

**Recall**: Recall measures the proportion of correctly predicted spam emails among all actual spam emails.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

In our example:

$$\text{Recall} = \frac{85}{85+15} = \frac{85}{100} = 0.85$$

So, the recall of our model for predicting spam emails is 0.85 or 85%.

These metrics provide insights into the performance of our model. A high precision indicates that when the model predicts an email as spam, it is likely to be correct. A high recall indicates that the model is good at capturing most of the actual spam emails. By using confusion matrices and calculating precision and recall, we can better understand the effectiveness of our classification model in practical scenarios.

## 2.3 Conclusion

In Chapter 2, we laid the groundwork for our research by introducing basic concepts in ML and digging into DL, the focus of our study. We went over supervised and unsupervised learning in depth, explaining their roles in categorization and pattern recognition. We also looked into two popular classification techniques, SVM and Random Forest. DL was introduced, with its autonomous feature learning capabilities, providing the framework for our exploration into DL architectures such as AlexNet, EfficientNetB0, ResNet50, Xception, DenseNet121, GoogleNet, Inception V3, VGG16, VGG19 and ViT Transformer. We introduced transfer learning technique and performance evaluation methodologies to ensure robust model assessment. These introductory parts have prepared us for a thorough investigation of DL's performance and interpretability in following chapters, which will provide vital insights into its practical uses and potential limitations.

# Chapter 3

# Literature on Explainability

Particularly in the area of brain tumor diagnosis using MRI scan images, the incorporation of DL models in medical image analysis has proven significant success in assisting diagnostics. The "black-box" nature of these models, which makes them opaque, upraise questions about their dependability and the ability to grasp how they make decisions. The scientific community has responded by aggressively investigating methods to measure the interpretability of DL models in medical imaging, with a focus on the detection of brain tumors. In order to evaluate the visual interpretability of DL models for brain tumor diagnosis using MRI scan pictures/images, this literature on explainability attempts to present an overview of major concepts, approaches, and findings in the field.

### 3.0.1 Interpretability Techniques in Medical Imaging

The decision-making processes of DL models in medical imaging are now being dissected by interpretability techniques, which also provide insights into how the algorithms make their predictions. The transparency and reliability of AI-driven medical diagnoses are increased by these techniques, which make it easier to identify pertinent features and areas within the images that contribute to the model's choices. Techniques for interpretability have been developed to reveal how DL models produce predictions. LIME [29] ,Grad-CAM [30], Integrated Gradients [31], and SHAP [32] have eminent themselves from the rest of these methods by highlighting the image regions that are most important to the model's conclusion. For instance, Grad-CAM creates heatmaps that highlight significant image regions, assisting doctors in comprehending the model's focal point.

### 3.0.2 Interpretability in Brain Tumor Diagnosis

Although DL models for diagnosing brain tumors have attained amazing levels of accuracy [31, 32], their intricate internal mechanisms raise questions about their dependability and generalizability. In order to solve this, academics have started looking into how interpretable these models are. To improve a deep learning model's transparency, [33] created a brand-new technique for viewing the tumor localization.

### 3.0.3 Quantifying Interpretability Metrics

It's difficult to gauge how well DL models can be interpreted. According to [34, 35], several criteria have been put out to evaluate consistency, localization accuracy, and sensitivity to perturbations. These measures are intended to offer a consistent framework for evaluating how closely the model's explanations match clinical expectations and actual tumor sites.

### 3.0.4 Integrating Interpretability in Clinical Practice

Medical practitioner's ability to make decisions may be improved by fusing interpretable DL models into clinical practice. Recent research [36] has shown how radiologists and doctors can use interpretable model explanations to validate the model's predictions and produce more accurate diagnoses. Radiologists participated in a study by [37, 38, 39] to assess the interpretability of DL models in medical imaging. The study used user studies to get expert opinion on the usefulness and clarity of the model explanations. The research underlined how crucial interpretability is to fostering communication among medical specialists.

## 3.1 Adaptive Path-Based Techniques

We explored the complexities of transfer learning models and path-oriented methods. This investigation is essential because it offers a thorough grasp of the theoretical underpinnings of the models we trained and the attribution techniques used to ensure explainability. Through providing in-depth analyses of these techniques, we want to provide clarification on the fundamental ideas that underpin how they work. This theoretical foundation is essential to understanding the subtleties of the models and their interpretability, which is consistent with our goal of developing a deeper grasp of the theoretical foundations of these approaches in addition to putting them into practice.

### 3.1.1 Grad-CAM and Grad-CAM++

Grad-CAM and Grad-CAM++ focus on visualizing the areas in the image that contribute significantly to the CNN's decision-making process, providing interpretability to the model's predictions. For Grad-CAM, let $A$ be a feature map of class $c$ and $Y^c$ the output that corresponds to class $c$. Its weights $\alpha_k^c$ at the $ij$-th position of the $k$-th feature map is expressed as:

$$\alpha_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial Y^c}{\partial A_{ij}^k}, \tag{3.1}$$

where $Z$ represents the feature map's size and $A_{ij}^k$ the activation of the unit in position $ij$ of the $k$-th feature map. In Grad-CAM++, the gradient weight $\alpha_{ij}^{kc}$ and $ReLU$ function that correspond to class $c$ are added, and the weight $w_k^c$ is expressed as:

$$w_k^c = \sum_i \sum_j ReLU\left(\frac{\partial Y^c}{\partial A_{ij}^k}\right) \alpha_{ij}^{kc}. \tag{3.2}$$

The localization heatmap $L$ of class $c$ in position $ij$ is expressed as:

$$L_{ij}^c = \sum_k w_k^c A_{ij}^k. \tag{3.3}$$

$ReLU$ is applied to enhance the relevance of positive gradients. These Equations (3.1)–(3.3) collectively represent the process of generating a heatmap that highlights the important regions in the input image for making predictions related to class $c$.

### 3.1.2 Integrated Gradient (IG)

Using this method, an attribution map is produced that shows the image parts that are important for the categorization choice. The output $h_c(y)$ represents the confidence score for predicting class $c$ given a classifier $h$, input $y$, and class $c$. To compute Integrated Gradients (IG), we performed a line integral between a reference point $y'$ and an image $y$ in the vector field generated by the gradient of $h_c(y)$ with respect to the input space. This vector field helps IG determine the importance or attribution for each feature, such as a pixel in an image. Formally, IG is defined as follows for each feature $i$:

$$I_i^{\mathrm{IG}}(y) = \int_0^1 \frac{\partial h_c(\gamma^{\mathrm{IG}}(\alpha))\partial \gamma_i^{\mathrm{IG}}(\alpha)}{\partial \gamma_i^{\mathrm{IG}}(\alpha)\partial \alpha} d\alpha, \tag{3.4}$$

where $\gamma^{\mathrm{IG}}(\alpha), \alpha \in [0,1]$ is the parametric function representing the path from $y'$ to $y$, with $\gamma^{\mathrm{IG}}(0) = y'$ and $\gamma^{\mathrm{IG}}(1) = y$. Specifically, $\gamma^{\mathrm{IG}}$ is a straight line connecting $y'$ and $y$.

### 3.1.3 Saliency Mapping

A method called saliency mapping uses an analysis of each pixel's impact on the classification score to determine how salient it is in an input image. Saliency maps show the visual regions that influence the classification choice with a linear scoring model [44]:

$$S_c(I) = w_c^T I + b_c, \tag{3.5}$$

where $b_c$ is the bias for class $c$, $w_c$ is the vector of weights, and $I$ is a single-dimensional vectorized description of the image's pixels. It is easy to see that in such a situation, the related pixels of $I$ are important based on the importance of the components of $w_c$. We cannot simply use this insight since $S_c(I)$ is a non-linear function of the image in deep convolutional networks. However, by evaluating the first-order Taylor expansion, we can roughly estimate the class score function with a linear function in the vicinity of a given image $I_0$:

$$S_c(I) \approx w^T I + b, \tag{3.6}$$

where $w$ is the derivative of $S_c$ with respect to the image $I$ at the point (image) $I_0$:

$$w = \left.\frac{\partial S_c}{\partial I}\right|_{I_0}. \tag{3.7}$$

Equation (3.7) can also be used to calculate the image-specific class saliency. In this case, the derivative's magnitude shows which pixels require modification.

## 3.2 Challenges and Future Directions

Although there has been advancement in evaluating interpretability, difficulties still exist in applying approaches to medical images, which frequently contain complex structures and diseased states. There are still pending questions regarding the therapeutic applicability of interpretable models and the relationship between interpretability and clinical results.

## 3.3 Conclusion

In conclusion, the research on explainability and interpretability techniques in medical imaging, strikingly in the factors of brain tumor diagnosis, shows an increasing awareness of the significance of transparency and trustworthiness in DL models. To gather knowledge about the inner workings of these models and give clinicians results that are understandable, researchers are increasingly creating and utilizing techniques like Grad-Cam, Grad-Cam++, Integrated Gradient (IG) and saliency mapping.

# Chapter 4

# Materials and Methodology

Chapter 4, forms the essential framework of our research. In Section 4.1, we fully explore our dataset, providing insights into its origins, characteristics, and relevance. Section 4.2 then takes a closer look at our experimental setup, offering transparency into the tools, algorithms, and methodologies we are using in this study. This chapter serves as the bridge between the theoretical support of our study and its practical implementation.

## 4.1 Dataset

For easy access and reference, the dataset is currently available on Kaggle[1]. This brain tumor dataset comprises 3264 2D slices of T1-weighted contrast-enhanced images, encompassing three distinct types of brain tumors—glioma, meningioma, and pituitary tumors—along with images of a healthy brain. We allocated 90% of the dataset for training and validation, reserving the

[1] https://www.kaggle.com/datasets/sartajbhuvaji/brain-tumor-classification-mri/data

remaining 10% for testing. We applied various data augmentation techniques to further enrich the training and validation subsets. The augmented subsets were subdivided into 90% for training and 10% for validation. The original images, initially sized at $299 \times 299$, were resized to $150 \times 150$ to enhance computational efficiency and facilitate model training. Figure 4.1 visually presents samples from each of the four classes within the dataset, and Figure 4.2 illustrates the proportion of each of the four classes in the dataset. Gliomas are tumors derived from glial cells and can manifest as either benign or malignant. Among them, glioblastoma multiforme stands out as a particularly aggressive variant, posing significant challenges in terms of therapeutic intervention [3]. Pituitary tumors arise in the pituitary gland; these tumors can disrupt hormonal balance. They may present as growths that secrete hormones or as non-functioning growths. Common sub-types include prolactinomas and growth-hormone-secreting tumors, each with its distinct clinical implications. Meningiomas are generally benign, slow-growing tumors originating from the meninges. The symptoms associated with meningiomas vary based on the size and location of the tumor, making their clinical presentation diverse and often dependent on individual cases.

## 4.2 Experimental Setup

This project proposes a comprehensive framework for brain tumor classification while integrating the model's explainability. This approach comprises two phases: phase (a) for classification and phase (b) for explainability, as illustrated in Figure 4.3.

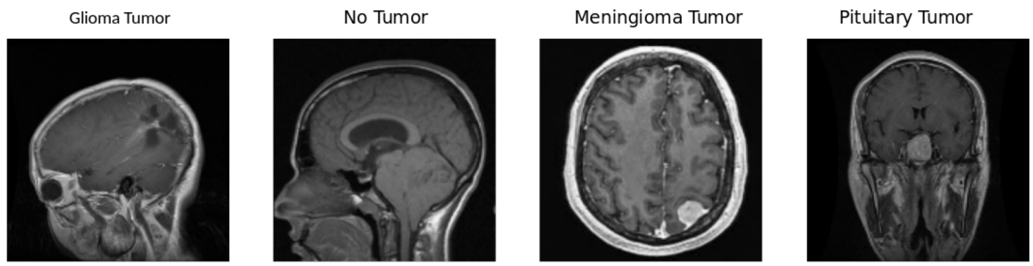


Figure 4.1: Sample of each image in the dataset. A Glioma Tumor image typically exhibits abnormal growth in the brain, indicating potential malignancy. No Tumor images represent a healthy state without any abnormal growth or lesions. Meningioma Tumor images showcase tumors arising from the meninges, the protective layers around the brain, and the spinal cord. Pituitary Tumor images depict tumors in the pituitary gland, influencing hormone regulation and potentially affecting various bodily functions.

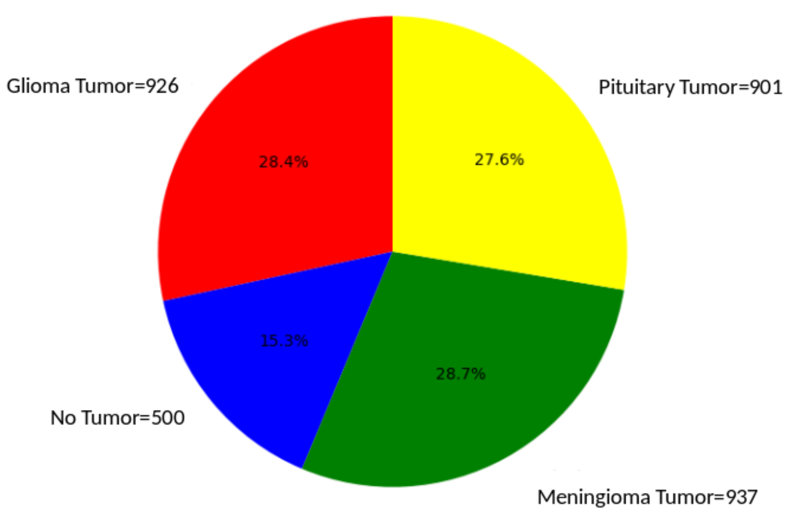


Figure 4.2: Brain tumor distribution. This pie chart effectively visualizes the relative proportions of different brain tumor types, offering a clear and concise representation of the distribution within the studied sample. The dataset comprises 926 MRI images of Glioma Tumors, 500 images with No Tumors, 901 images featuring Pituitary Tumors, and 937 images showing Meningioma Tumors.

Phase (a): This phase integrates a CNN model with TL for the classification of brain tumor types based on MRI data. In this approach, the acquired features of a pre-trained CNN model serve as an initial foundation, proving particularly advantageous in scenarios involving sparsely labeled data. The data are fed into a CNN, which subsequently processes the data through convolutional layers to capture intricate patterns and spatial hierarchies within the MRI images. Following this, the pooling layer is employed to downsample and reduce the feature space, optimizing computational efficiency. Progressing along the activation path, the dense layer plays a pivotal role in transforming high-level features for effective classification. Finally, the model makes decisions about tumor types based on the combination of these learned features. When the decision is made, a medical expert becomes curious and seeks to understand how the model makes decisions based on their expertise.

Phase (b): This phase uses explainability techniques, including Grad-CAM, Grad-CAM++, IG, and Saliency Mapping. The explanation aims to shed light on how the CNN model arrives at its classifications, providing valuable insights to the medical expert. Grad-CAM and Grad-CAM++ offer the visualization of crucial regions in the MRI images that contribute to the model's decisions. IG provides a comprehensive understanding of feature importance by perturbing input features, while Saliency Mapping highlights salient features influencing the classification. Together, these explainability techniques bridge the gap between the model's predictions and human interpretability, fostering trust and comprehension in the application of DL models to medical imaging. After the model is explained, the focus shifts smoothly to the part where medical experts take over and make sense of it. The explained visualizations and insights provided by techniques like Grad-CAM, Grad-CAM++, IG, and Saliency Mapping serve as a bridge between the complex nature of DL classifications and the expertise of medical experts.

### 4.2.1 Training, regularization, and testing

This section elaborates on the model training process, hyperparameter tuning, and model testing, with a specific emphasis on the computational infrastructure used. As discussed in Section 4.1, 90% of the dataset was reserved for training and validation. This 90 % dataset was then augmented and shuffled to ensure randomness before splitting into 90% for training and 10% for validation. The hyperparameters, such as the number of layers, filter sizes, and dropout rates, were optimized to enhance the model's predictive capabilities. This involved a systematic exploration of different hyperparameter combinations using Grid Search. It took several days to run the Grid Search for hyperparameter tuning. Throughout this process, we explored various combinations of hyperparameters, such as learning rates ranging from 0.001 to 0.1, batch sizes from 16 to 32, epochs from 10 to 150, as well as filter sizes and the number of filters. After completing the search, the best combination of hyperparameters was found to be a learning rate of 0.001, a batch size of 32, and 10 epochs. We used the categorical cross-entropy as a loss function with the Adam optimizer. The details of these hyperparameters are provided in Table 4.1. Figure 4.4 depicts the training and validation losses, as well as the training and validation accuracies for the top-performing models. It is evident from this figure that the models did not exhibit further

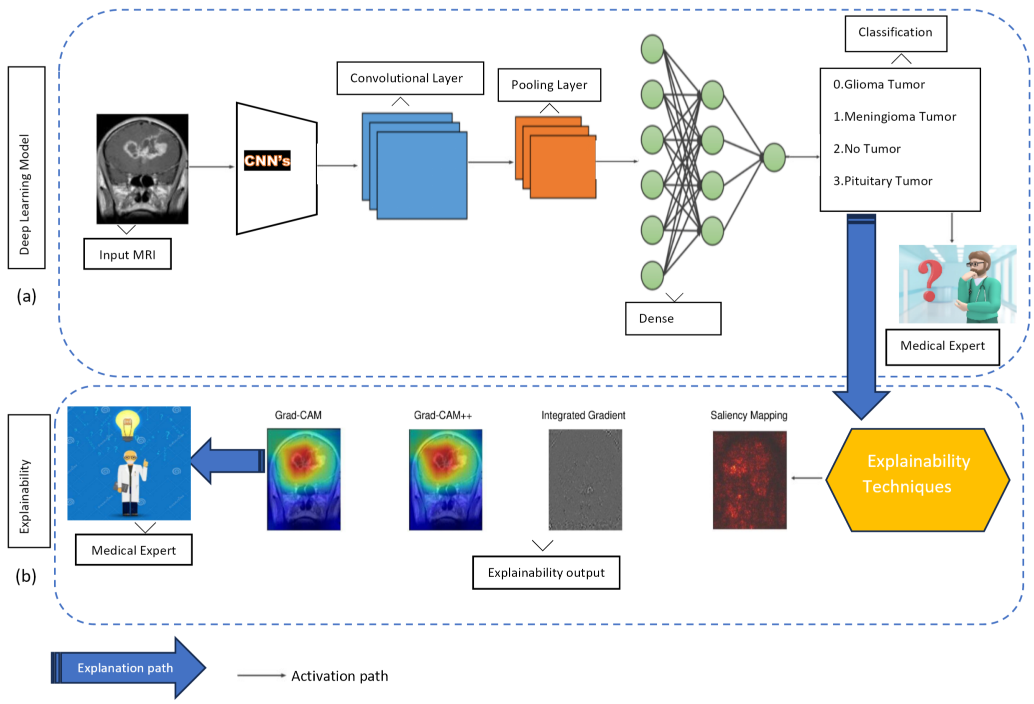


Figure 4.3: Comprehensive Framework for Brain Tumor Classification: Integrating Classification Accuracy and Model Explainability. (a) In the initial phase of our study, we focused on classifying different types of tumors. However, despite achieving classification results, we found ourselves in the dark about why the models made specific decisions. (b) To address this gap, we transitioned to the next stage of our investigation, delving into explainability methods. In this phase (b), we aim to shed light on the decision-making processes of our models. By employing explainability tools, we seek to unravel the factors and features that influenced the model's classifications. This shift allows us to not only categorize tumors accurately but also understand the underlying rationale behind the model's predictions, contributing to a more informed and interpretable outcome.

learning beyond epoch 10, as indicated by the convergence of the training and validation losses. Hence, we preferred to stop training at epoch 10. The testing phase aimed to evaluate the trained model's generalization to new, unseen data. The 10% test dataset as discussed in Section 4.1, consisting of 327 images, was used for this purpose. Metrics such as accuracy, precision, recall, and the so-called F1 score were computed to assess the model's performance. The model training and hyperparameter tuning of our pre-trained models demanded substantial computational resources, and the efficiency of the process hinged on the specifications of our Dell computer, featuring an Intel Core i7 processor. The use of a powerful GPU, specifically the P100 model, played a pivotal role in accelerating the training speed. With 32 GB of system RAM, our system efficiently handled the computational load, facilitated by the Kaggle kernel environment. The implementation was carried out in *Python* version 3.9.7 and *TensorFlow* version 2.12.0.

Table 4.1: Training Hyperparameters

| Hyperparameter | Setting |
|---|---|
| Batch size | 32 |
| Learning rate | 0.001 |
| Epochs | 10 |
| Training and validation split | 90% |
| Test split | 10% |
| Optimizer | Adam |
| Input size | $150 \times 150$ pixels |
| Loss function | Categorical cross-entropy |

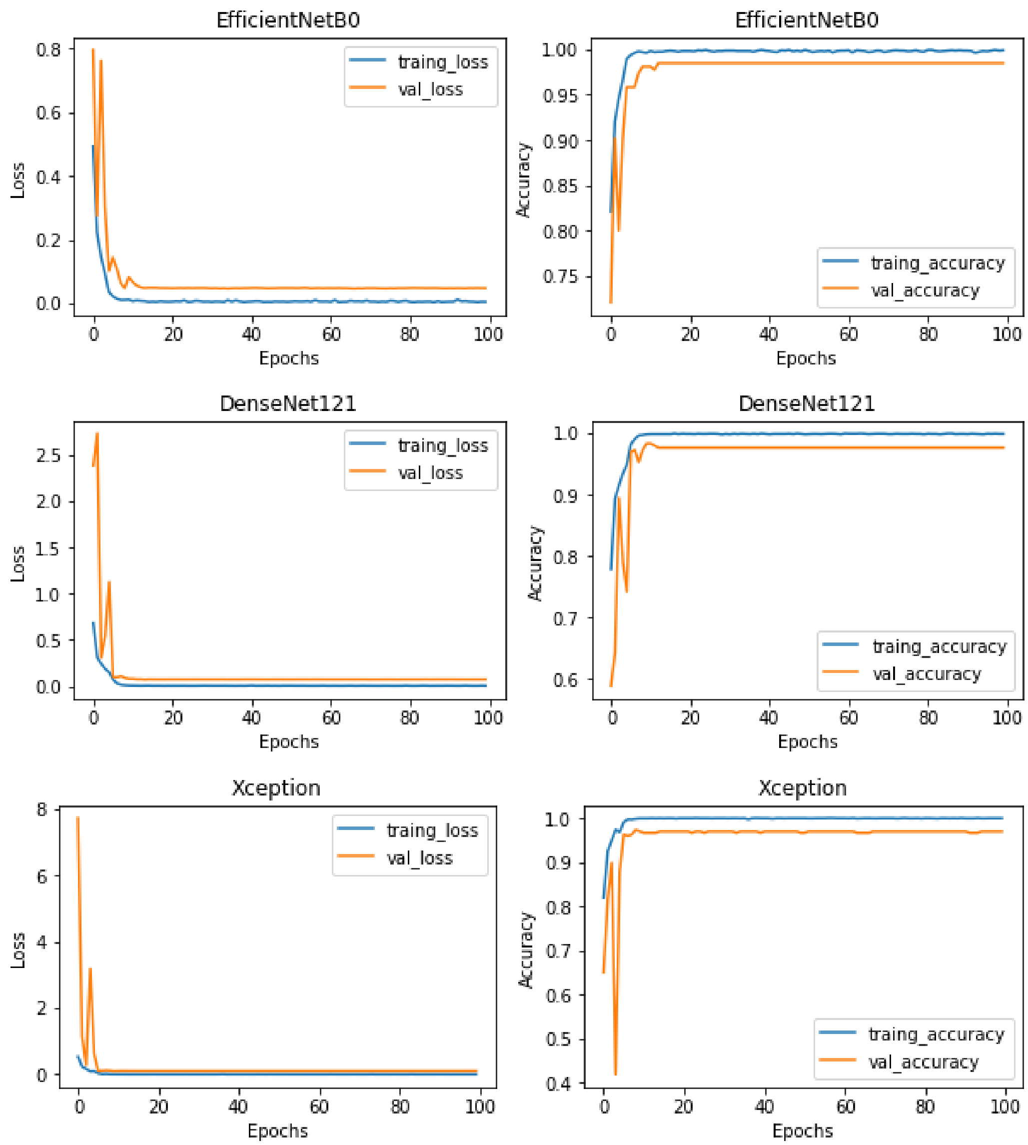


Figure 4.4: EfficientNetB0, DenseNet121, and Xception, during training and validation. The plots illustrate the training and validation loss, as well as the training and validation accuracy over multiple epochs.

# Chapter 5

# Results and Discussions

In this section, we present the outcomes of our study, offering a comprehensive overview of the key findings and their interpretation. Through this detailed analysis, we aim to provide a thorough understanding of the performance, interpretability, and decision-making processes inherent in the deep learning models employed in this work. Our contributions extend to the broader field of model interpretability, enhancing the collective knowledge within the research community.

## 5.1 Classification results

The detailed training outcomes are presented in Table 5.1 and visually represented in Figure 5.1. Moving on to the test results, a summary is provided in Table 5.2, and Figure 5.2 visually highlights the superior performance of the best model. This section also undertakes a thorough analysis and interpretation of the confusion matrix derived from the classification. A detailed examination follows, shedding light on the interpretability results. Specifically, results for adaptive path-based techniques, such as Grad-CAM,

Table 5.1: Training Results

| Model Name | Parameters | Training Accuracy | Loss |
|---|---|---|---|
| AlexNet | 61.9 M | 0.8763 | 0.3233 |
| DenseNet121 | 8.1 M | 0.9986 | 0.0057 |
| EfficientNetB0 | 5.3 M | 0.9991 | 0.0042 |
| GoogLeNet | 11.2 M | 0.9997 | 0.0027 |
| Inception V3 | 23.9 M | 0.9989 | 0.0084 |
| ResNet50 | 25.6 M | 0.9991 | 0.0044 |
| VGG16 | 138.4 M | 0.8698 | 0.4011 |
| VGG19 | 143.7 M | 0.8570 | 0.3953 |
| Vision Transformer | 86 M | 0.7484 | 0.5115 |
| Xception | 22.9 M | 1.0000 | 0.0021 |

Grad-CAM++, IG, and Saliency Mapping, are discussed in depth.

The training outcomes, as detailed in Table 5.1, offer insights into the model's ability to learn from the training data. High training accuracy and low training loss often signify successful training, yet these metrics may not necessarily ensure performance on the test data. Among the models investigated in this study, DenseNet121, EfficientNetB0, GoogLeNet, Inception V3, ResNet50, and Xception stood out with an acceptable training accuracy, ranging from 99.86% to 100%. These high accuracy scores indicate that these models have effectively learned the statistical regularities present in the training data. Furthermore, the associated training loss values were remarkably low, showcasing the models' efficiency in minimizing errors during the training phase. It is imperative to note that achieving high training accuracy does not necessarily guarantee superior performance on unseen datasets, emphasizing the importance of the comprehensive evaluation of the test data for a more robust assessment of model generalization.

In contrast, models with notably lower training accuracy, spanning from 74.84% to 87.63%, such as AlexNet, VGG16, VGG19, and Vision Transformer, while still demonstrating commendable performance on the training data, exhibited comparatively higher training loss values. This suggests a slightly higher level of modeling error during the learning process for these models. The inferior performance of VGG16, VGG19, ViT Transformer, and AlexNet may be due to a combination of huge parameter counts and excessive model complexity, which may not be adequately aligned with the task's features. To increase the generalization capacity of these models, regularization strategies like dropout or batch normalization may need to be further refined or optimized, as well as further data augmentation.

The classification results, as detailed in Table 5.2, provide valuable insights into the model's capability to classify the test data; it became evident that EfficientNetB0 emerged as the superior model among all those considered in this study. Upon closer examination of the results, it was apparent that some of the models may not be suited for this specific task, with some requiring more computational resources due to higher parameter counts, as observed in VGG16 and VGG19.

Figure 5.2 compares the 10 deep learning models used in this study, highlighting each model's performance across key criteria such as accuracy, precision, recall, and F1 score percentages. This graphical representation offers a concise summary of these models' effectiveness in classifying brain tumors from MRI images, enabling a quick and informative comparison. EfficientNetB0,

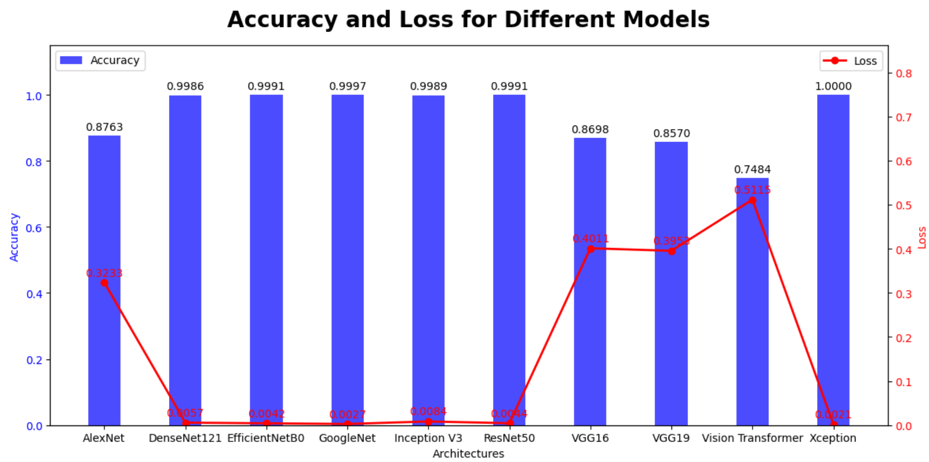


Figure 5.1: Training accuracy and loss are the two main keys that are shown in this figure, which shows the training outcomes of several models. A distinct DL model is represented by each bar in the plot, and the height of the bar indicates the model's training accuracy. Additionally, the loss values for each model are displayed as a line plot superimposed on the same graph.

exhibiting the best performance, showed great promise and warrants further consideration.

Figure 5.3 displays the confusion matrix for the EfficientNetB0 model. The diagonal elements represent samples that were accurately predicted. Out of the total of 327 samples, 321 were predicted correctly, resulting in an overall accuracy of 98%. The element in row 1 and column 2, a value of 2, indicates that EfficientNetB0 incorrectly learned the classification boundary between classes 1 and 2. This implies that the model confused data initially belonging to class 2 with class 1. Conversely, the element in row 4 and column 1, a value of 0, signifies that the classification boundary between classes 1 and 4 was correctly learned by EfficientNetB0, and the model did not confuse data initially belonging to class 4 with class 1.

Table 5.2: Test dataset classification results where various metrics are considered for each model.

| Model Name | Accuracy % | Precision % | Recall % | F1 Score % |
|---|---|---|---|---|
| AlexNet | 78 | 80 | 77 | 77 |
| DenseNet121 | 97 | 97 | 97 | 97 |
| EfficientNetB0 | 98 | 98 | 98 | 98 |
| GoogleNet | 91 | 93 | 92 | 92 |
| Inception V3 | 96 | 97 | 96 | 96 |
| ResNet50 | 95 | 96 | 96 | 96 |
| VGG16 | 85 | 85 | 86 | 85 |
| VGG19 | 85 | 85 | 85 | 85 |
| ViT Transformer | 70 | 72 | 72 | 70 |
| Xception | 96 | 97 | 96 | 96 |

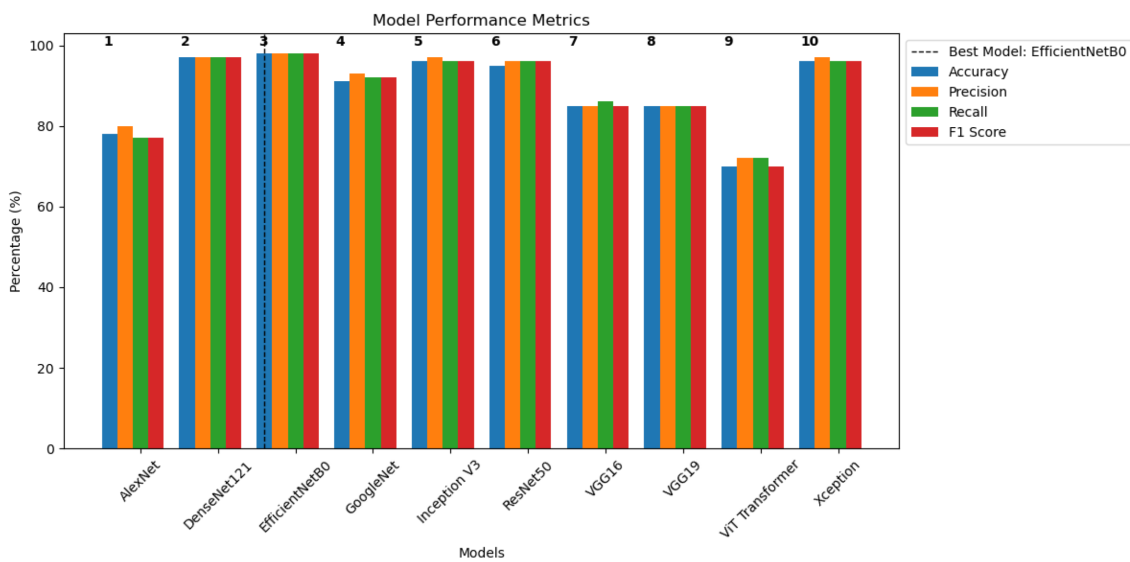


Figure 5.2: Visualizing the comparison of different DL models to provide a clear overview of their performance. The main performance indicators used to assess the effectiveness of various models are highlighted in the legend. The top model is named as EfficientNetB0 due to its excellent results in accuracy, precision, recall, and F1 score, which demonstrate its ability to provide well-balanced and accurate predictions on the provided test dataset.


Figure 5.3: EfficientNetB0 confusion matrix. The matrix systematically breaks down the model's predictions, highlighting instances of true positives (correctly identified cases), true negatives (correctly rejected cases), false positives (incorrectly identified cases), and false negatives (missed cases) for each tumor class.

For the model DenseNet121 in Figure 5.4, the confusion matrix shows that, out of the total of 327 samples, 320 were accurately predicted, resulting in an overall accuracy of 97%. The value 3 in row 1 and column 2 indicates that the classification boundary between classes 1 and 2 was incorrectly learned by DenseNet121, suggesting that the model confused data initially belonging to class 2 with class 1. Similarly, the presence of 0 in row 2 and column 1 implies that the classification boundary between classes 2 and 1 was correctly learned by DenseNet121, and the model did not confuse data initially belonging to class 1 with class 2. Figure 5.5 depicts the confusion matrix for the model Xception. Out of the total of 327 samples, 315 were accurately predicted, resulting in an overall accuracy of 96%. The value of 1 in row 1 and column 2 indicates that the classification boundary


Figure 5.4: DenseNet121 confusion matrix.

between classes 1 and 2 was incorrectly learned by Xception, suggesting that the model confused data initially belonging to class 2 with class 1. Additionally, the presence of 0 in row 1 and column 4 signifies that the classification boundary between classes 1 and 4 was correctly learned by Xception, and the model did not confuse data initially belonging to class 4 with class 1.

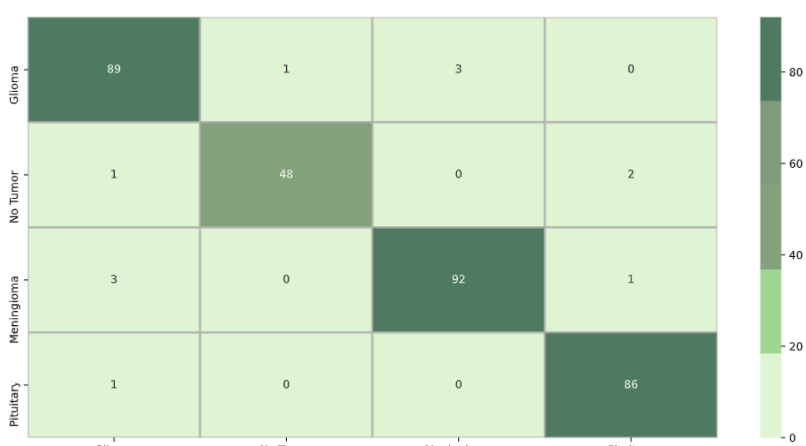


Figure 5.5: Xception confusion matrix.

## 5.2 Interpretability results

In Figures 5.6–5.8, we present a detailed look into the explainability results of our top-performing models: EfficientNetB0, DenseNet121, and Xception. Figure 5.6 provides insights into the interpretability of EfficientNetB0, highlighting crucial image regions using techniques such as Grad-CAM, Grad-CAM++, IGs, and Saliency Mapping. Moving to Figure 5.7, we explore the explainability of DenseNet121, our second-best model, uncovering the significant features influencing its predictions. Figure 5.8 reveals the interpretability degrees of Xception, our third-best model, showcasing the impact of various image regions on the classification decisions. These visualizations offer a transparent view into the decision-making processes of our models, facilitating understanding and trust.

Based on our obtained results, both the Grad-CAM and Grad-CAM++ methods demonstrated similar outcomes, offering visual explanations for the decision-making processes of EfficientNetB0, DenseNet121, and Xception in predicting brain tumors. These methods accurately pinpointed the exact location of the tumor, and the visualizations generated by both approaches closely aligned, suggesting a consistent portrayal of the crucial regions or features influencing the model's predictions. However, it is essential to note that IG faced challenges in precisely pinpointing the tumor's location, and Saliency Mapping exhibited some noise. Despite these challenges, both Grad-CAM and Grad-CAM++ consistently provided excellent visual explanations for the decision processes of EfficientNetB0, DenseNet121, and Xception, significantly enhancing our understanding of their predictive mechanisms.

In the case of "no tumor", the presence of a red cloud color, though less intense and more centralized compared to images with tumors, indicates that features associated with the absence of tumors are faint and centered in the image. This insight provides clarity on how the deep learning model interprets images without tumors.

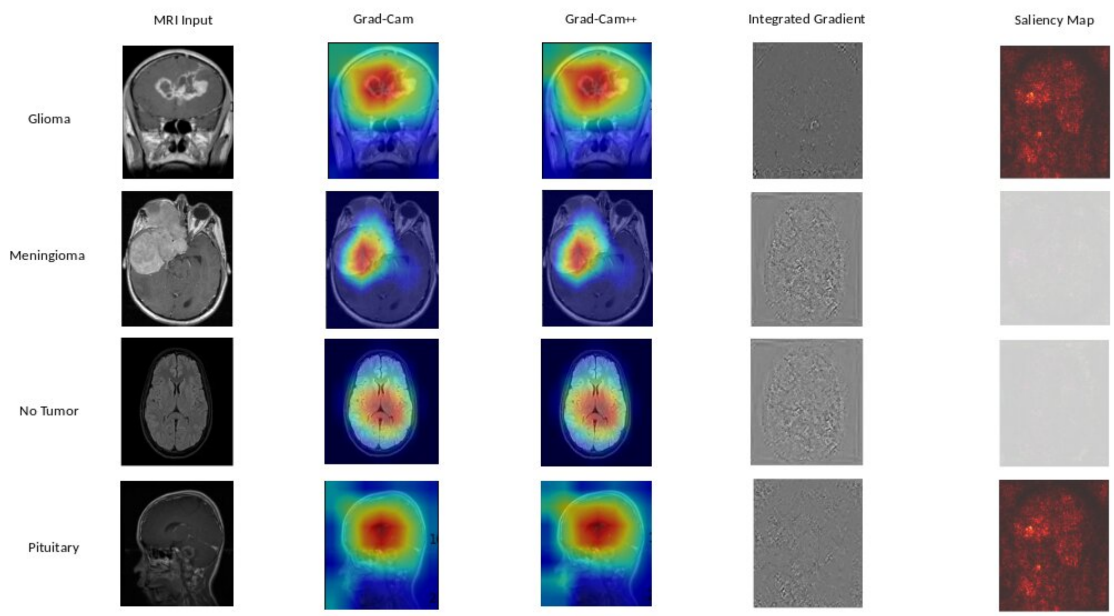


Figure 5.6: EfficientNetB0 explainability: We used a variety of explainability approaches, such as Grad-CAM, Grad-CAM++, IG, and Saliency Mapping, in our evaluation of EfficientNetB0 for brain tumor classification. These techniques played an important role in assisting in identifying the specific regions in MRI scan images that corresponded to the tumor types such as glioma, meningioma, no tumor, and pituitary. By using these techniques, we were able to determine the critical locations for the categorization of each tumor type and obtain important insights into EfficientNetB0's decision-making process.

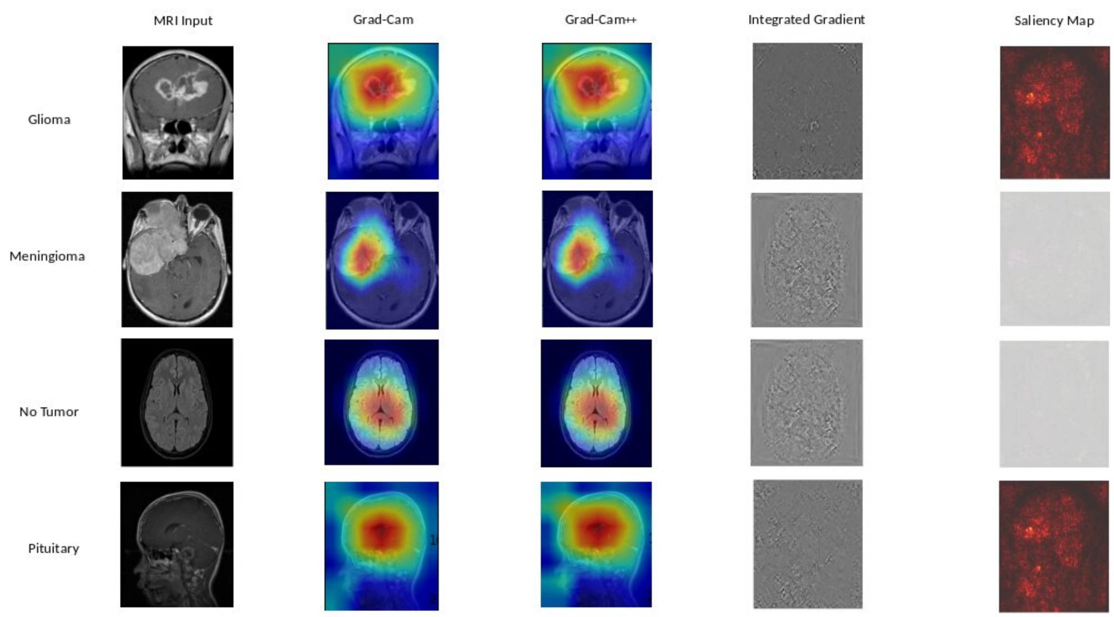


Figure 5.7: DenseNet121 explainability: We used a variety of explainability approaches, such as Grad-CAM, Grad-CAM++, IG, and Saliency Mapping, in our evaluation of DenseNet121 for brain tumor classification. These techniques played an important role in assisting in identifying the specific regions in MRI scan images that corresponded to the tumor types such as glioma, meningioma, no tumor, and pituitary. By using these techniques, we were able to determine the critical locations for the categorization of each tumor type and obtain important insights into DenseNet121's decision-making process.

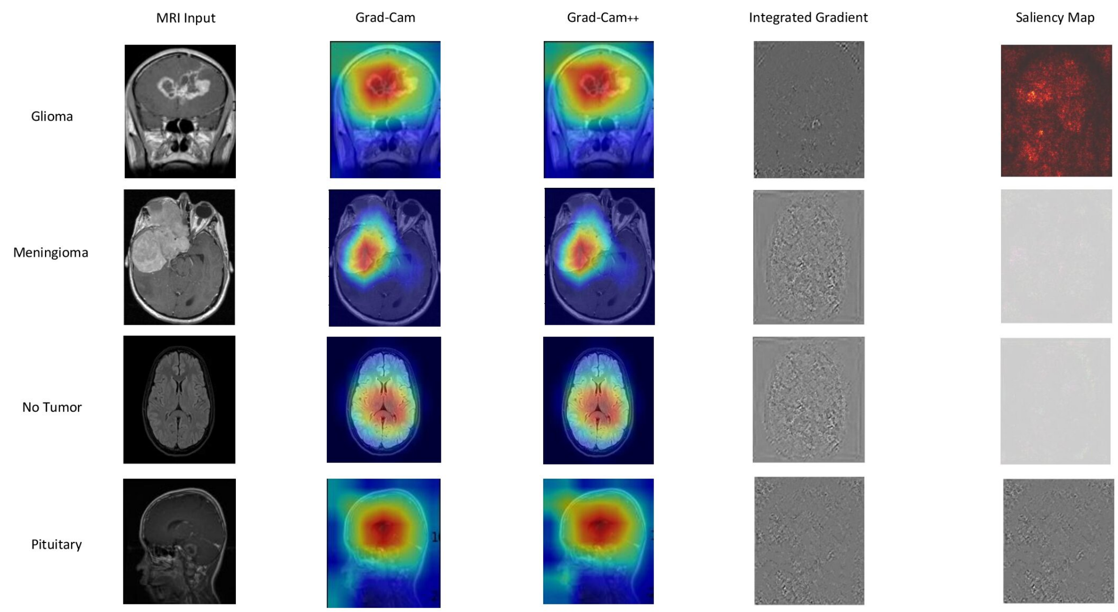


Figure 5.8: Xception explainability: We used a variety of explainability approaches, such as Grad-CAM, Grad-CAM++, IG, and Saliency Mapping, in our evaluation of Xception for brain tumor classification. These techniques played an important role in assisting in identifying the specific regions in MRI scan images that corresponded to the tumor types such as glioma, meningioma, no tumor, and pituitary. By using these techniques, we were able to determine the critical locations for the categorization of each tumor type and obtain important insights into Xception's decision-making process.

### 5.2.1 Discussion

The findings of evaluating the performance of different DL models provide intriguing new information on the relationship between total accuracy and model complexity, which is reflected by the number of parameters shown in Table 5.1. EfficientNetB0, DenseNet121, and Xception showed impressive accuracy values of 97-98% with much fewer parameters, indicating how well they use information for the tasks at hand. On the other hand, the accuracy was about 85% for VGG16 and VGG19, which have importantly greater parameter counts of more than 138 million. There is a clear exchange between model accuracy and complexity, highlighting the necessity of striking a balance. Interestingly, models with a reasonable amount of parameters, such as AlexNet, ResNet50, GoogLeNet, and Xception, achieved competitive accuracy in the 80-96% range, striking a medium ground. Even with a low number of parameters, the ViT Transformer showed greater accuracy variability, highlighting the impact of the architectural design on model performance.

Grad-CAM, Grad-CAM++, IG, and Saliency Mapping showed important areas that matched the glioma, meningioma, and pituitary classifications. What stood out was the remarkable similarity in the outcomes between the Grad-CAM and Grad-CAM++ methods, precisely pinpointing tumor locations. This consistency paints a clear picture of the essential regions influencing predictions across all three models; see Figures 5.6–5.8. These visualizations played a crucial role, not just in making our models transparent, but also in fostering a deeper understanding and instilling trust in how the decisions were made.

# Conclusion

In essence, the objective of this investigation was to assess the proficiency of diverse DL models in categorizing brain tumors. Following a series of comprehensive experiments and detailed analysis, it became apparent that these models exhibited varying degrees of competence for the assigned task. Models such as DenseNet121, EfficientNetB0, ResNet50 GoogLeNet, and Inception V3 distinguished themselves as top performers with almost flawless levels of accuracy, precision, recall, and F1 scores. For detailed classification results, refer to Table 5.2. Conversely, AlexNet and the innovative ViT Transformer, a recent contender in the field, displayed potential, but fell behind in terms of accuracy and achieving an optimal equilibrium between precision and recall. This research accentuates the significance of carefully selecting the most suitable DL model that aligns with the specific requirements of the application. It further underscores how advancements in neural network architectures, exemplified by the ViT Transformer, persist in shaping the field of DL and computer vision, presenting captivating prospects for future advancements.

In summary, both Grad-CAM and Grad-CAM++ consistently provided a more acceptable insight into model interpretability compared to other methods tested in our study. Put simply, these methods precisely revealed the location of tumors, significantly enhancing our understanding of how DL models make decisions in classifying brain tumors. Therefore, it can be concluded that Grad-CAM's and Grad-CAM++'s heatmaps have improved our interpretative accuracy, playing a pivotal role in refining our understanding of DL model decision-making processes. These methodologies have been instrumental in enhancing the precision of our interpretations. This study contributes to selecting the correct DL model for brain tumor classification tasks while shedding light on ongoing challenges in making these models transparent and interpretable. However, further extensive work could be carried out to compare correlation measures and feature localization precision among the different studied models.

# References

[1] D. Bau, B. Zhou, A. Khosla, A. Oliva, A. Torralba, Network dissection: Quantifying interpretability of deep visual representations, in: Proceedings of the IEEE conference on computer vision and pattern recognition,2017, pp. 6541–6549.

[2] Abiwinanda, N., Hanif, M., Hesaputra, S.T., Handayani, A. and Mengko, T.R., 2019. Brain tumor classification using convolutional neural network. In World Congress on Medical Physics and Biomedical Engineering 2018: June 3-8, 2018, Prague, Czech Republic (Vol. 1) (pp. 183-189). Springer Singapore.

[3] Ghazanfar Latif et al. "Multiclass brain Glioma tumor classification using block-based 3D Wavelet features of MR images." In: 2017 4th International Conference on Electrical and Electronic Engineering (ICEEE). IEEE. 2017, pp. 333–337.

[4] Fukushima, K., 1980. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biological cybernetics, 36(4), pp.193-202.

[5] Brima, Y. and Atemkeng, M., 2022. What do Deep Neural Networks Learn in Medical Images?. arXiv preprint arXiv:2208.00953.

[6] Potok, T.E., Schuman, C., Young, S., Patton, R., Spedalieri, F., Liu, J., Yao, K.T., Rose, G. and Chakma, G., 2018. A study of complex deep learning networks on high-performance, neuromorphic, and quantum computers. ACM Journal on Emerging Technologies in Computing Systems (JETC), 14(2), pp.1-21.

[7] Krizhevsky, A., Sutskever, I. and Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25.

[8] [Huang et al., 2017] Gao Huang, Zhuang Liu, Kilian Q. Weinberger, and Laurens van der Maaten. Densely connected convolutional networks. In CVPR, 2017.

[9] He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[10] Golriz Khatami, S., and Moghaddam, H. A. (2013). Support Vector Machines in Medical Imaging: A Review. Journal of Medical Signals and Sensors, 3(2), 62-76.

[11] Cortes, C., and Vapnik, V. (1995). Support-vector networks. Machine Learning, 20(3), 273-297.

[12] Zhang, Z., Chen, P., and Liang, X. (2018). Brain tumor detection using support vector machine on magnetic resonance images. Journal of Healthcare Engineering, 2018.

[13] Simonyan et al., 2013. Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: visualising image classification models and saliency maps. In arXiv:1312.6034, 2013.

[14] Springenberg et al., 2015. Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: the all convolutional net. ICLR workshop, 2015.

[15] Olah et al., 2017. Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. Distill, 2017.

[16] Dosovitskiy and Brox, 2016. Alexey Dosovitskiy and Thomas Brox. Inverting visual representations with convolutional networks. In CVPR, 2016.

[17] Nguyen, A., Clune, J., Bengio, Y., Dosovitskiy, A. and Yosinski, J., 2017. Plug and play generative networks: Conditional iterative generation of images in latent space. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4467-4477).

[18] Fong and Vedaldi, 2017. Ruth C. Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In ICCV, 2017.

[19] Ribeiro et al., 2016. Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?" explaining the predictions of any classifier. In KDD, 2016. Sabour et al., 2017. Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. Dynamic routing between capsules. In NIPS, 2017. Selvaraju et al.,

[20] Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. (2017). Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. In ICCV.

[21] Lundberg, S. M., and Lee, S. I. (2017). A Unified Approach to Interpreting Model Predictions. In NeurIPS.

[22] Sheldon Mascarenhas; Mukul Agarwal. (2021). A comparison between the VGG16, VGG19 and ResNet50 architecture frameworks for image classification. 2021 International Conference on Disruptive Technologies for Multidisciplinary Research and Applications (CENTCON) .DOI:10.1109/CENTCON52345.2021.9687944.

[23] M. Böhle, F. Eitel, M. Weygandt, K. Ritter, Layer-wise relevance propagation for explaining deep neural network decisions in mri-based alzheimer's disease classification, Frontiers in aging neuroscience (2019) 194.

[24] Stefenon, S.F., Yow, K.C., Nied, A. and Meyer, L.H., 2022. Classification of distribution power grid structures using inception v3 deep neural network. Electrical Engineering, 104(6), pp.4557-4569.

[25] Kroll JA (2018) The fallacy of inscrutability. Philos Trans R Soc A 376(2133):20180084

[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in Proc. Adv. Neural Inf. Process. Syst., 2012, pp

[27] Tan, Y., Wu, P., Zhou, G., Li, Y. and Bai, B., 2020. Combining residual neural networks and feature pyramid networks to estimate poverty using multisource remote sensing data. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 13, pp.553-565.

[28] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S. and Uszkoreit, J., 2020. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929.

[29] Tan, M. and Le, Q., 2019, May. Efficientnet: Rethinking model scaling for convolutional neural networks. In International conference on machine learning (pp. 6105-6114). PMLR.

[30] DeAngelis, L.M., 2001. Brain tumors. New England journal of medicine, 344(2), pp.114-123.

[31] Havaei, M., Davy, A., Warde-Farley, D., Biard, A., Courville, A., Bengio, Y., and Pal, C. (2017). Brain tumor segmentation with Deep Neural Networks. Medical Image Analysis, 35, 18-31.

[32] Zhang, X., Yao, J., Wang, Y., and Zhang, L. (2019). Brain Tumor Classification Using Deep Learning Neural Networks. Journal of Healthcare Engineering, 2019, 1-8.

[33] Zhang, Z., Xie, Y., Xing, F., and McGough, M. (2021). Interpretability of deep learning models for medical image analysis: A survey. Medical Image Analysis, 101934.

[34] Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M., and Kim, B. (2018). Sanity Checks for Saliency Maps. In NeurIPS.

[35] Kim, B., Wattenberg, M., Gilmer, J., Cai, C., Wexler, J., Viegas, F., and Sayres, R. (2020). Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV). In ICML.

[36] Gao, J., Liang, J., Song, Z., and Xie, Y. (2019). Explainable AI in Radiology: Current Status and Future Directions. Bioengineering, 6(4), 94.

[37] Cheplygina, V., de Bruijne, M., Pluim, J. P., and Marchiori, E. (2021). A Survey of Uncertainty Quantification in Deep Learning for Medical Image Analysis. Medical Image Analysis, 101880.

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

[39] Zhang, Q., Wang, W. and Zhu, S.C., 2018, April. Examining CNN representations with respect to dataset bias. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 32, No. 1).

[40] Mahendran and Vedaldi, 2015. Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In CVPR, 2015.

[41] Krawczyk, B., and Woźniak, M. (2016). Random Forest ensemble with expert knowledge. Information Fusion, 27, 170-179.

[42] Chattopadhay, A., Sarkar, A., Howlader, P. and Balasubramanian, V.N., 2018, March. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In 2018 IEEE winter conference on applications of computer vision (WACV) (pp. 839-847). IEEE.

[43] Sundararajan, M., Taly, A. and Yan, Q., 2017, July. Axiomatic attribution for deep networks. In International conference on machine learning (pp. 3319-3328). PMLR.

[44] Simonyan, K., Vedaldi, A. and Zisserman, A., 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. arXiv preprint arXiv:1312.6034.