An Effective Model to Detect and Classify Brain Tumor using Deep Convolutional Neural Network

This thesis is submitted to the Department of Electronics and Telecommunication Engineering of Chittagong University of Engineering & Technology As a partial fulfillment of the requirement for the degree of Master of Science in Electronics and Telecommunication Engineering.

Submitted by

Shaila Shanjida

Student ID: 19METE031P

Supervised by

Dr. Md. Saiful Islam

Associate Professor

Department of Electronics and Telecommunication Engineering

Chittagong University of Engineering and Technology

MASTER OF SCIENCE IN ELECTRONICS AND TELECOMMUNICATION ENGINEERING

Chittagong University of Engineering & Technology

2024

APPROVAL

The thesis titled **"An Effective Model to Detect and Classify Brain Tumor using Deep Convolutional Neural Network"** submitted by Shaila Shanjida, ID-19METE031P, Session: 2019-20 has been accepted as satisfactory in partial fulfillment of the requirement of the degree of Master of Science (M.Sc.) in Electronics and Telecommunication Engineering on 5th September 2024.

<u>Board of Examiners</u>

1.	Dr. Md. Saiful Islam Associate Professor, Dept. of ETE, CUET	Chairman (Supervisor)
2.	Prof. Dr. Md. Azad Hossain Head, Dept. of ETE, CUET	Member (Ex-Officio)
3.	Prof. Dr. Md. Azad Hossain Professor, Dept. of ETE, CUET	Member (Internal)
4.	Prof. Dr. Quazi Delwar Hossain Professor, Dept. of EEE, CUET	Member (External)

Declaration

I hereby declare that the work contained in this thesis has not been previously submitted to meet requirements for an award at this or any other higher education institution. To the best of my knowledge and belief, the Thesis contains no material previously published or written by another person except where due reference is cited. Furthermore, the Thesis complies with PLAGIARISM and ACADEMIC INTEGRITY regulation of CUET.

Shaila Shanjida
19METE031P
Department of Electronics and Telecommunication Engineering (ETE)
Chittagong University of Engineering & Technology (CUET)

DEDICATION

Dedicated to

My beloved husband and parents

Without whom I would be nothing

List of Publications

- **Shaila Shanjida,** M. Mohiuddin, **M. S. Islam**, Jia Uddin. "A Novel Brain Tumor Detection and Classification Technique Using an effective Deep Parallel CNN", IEIE Transactions on Smart Processing and computing. **[Q4 journal, Scopus index]**

- **Shaila Shanjida,** M. Mohiuddin, and **M. S. Islam**. 2022. " A Novel Deep Learning Technique for Brain Tumor Detection and Classification using Parallel CNN with Support Vector Machine" Engineering Proceedings https://doi.org/10.3390/ecsa-11-13339 **[Jounal: Egineering Proceding, Scopus index, Citescore: 0.9]**

- **Shaila Shanjida**, **M. S. Islam** and M. Mohiuddin, "Hybrid model-based Brain Tumor Detection and Classification using Deep CNN-SVM," 2024 6th International Conference on Electrical Engineering and Information & Communication Technology (ICEEICT), Dhaka, Bangladesh, 2024, pp. 1467-1472, doi: 10.1109/ICEEICT62016.2024.10534376. **[Scopus index]**

- **Shaila Shanjida**, **M. S. Islam** and M. Mohiuddin, "MRI-Image based Brain Tumor Detection and Classification using CNN-KNN," 2022 IEEE IAS Global Conference on Emerging Technologies (GlobConET), Arad, Romania, 2022, pp. 900-905, doi: 10.1109/GlobConET53749.2022.9872168. **[Scopus index]**

Approval/Declaration by the Supervisor(s)

This is to certify that **Shaila Shanjida** has carried out this research work under my supervision, and that he has fulfilled the relevant Academic Ordinance of the Chittagong University of Engineering and Technology, so that he is qualified to submit the following Thesis in the application for the degree of Master of Science in ETE. Furthermore, the thesis complies with the PLAGIARISM and ACADEMIC INTEGRITY regulation of CUET.

Dr. Md. Saiful Islam

Associate Professor

Department of Electronics and Telecommunication Engineering

Chittagong University of Engineering & Technology

Acknowledgement

First of all, I would like to thank Allah Subhanahu Otaya'la for making things easier for me than I could ever bear. I would like to remember few very important persons without whom I would never be able to complete the thesis. In particular, I would like to acknowledge the contribution, inspiration and motivation from my honorable supervisor, Dr. Md. Saiful Islam, Associate Professor, Department of ETE, CUET. He has been a consistent guidance throughout my journey. I also would like to thank Prof. Dr. Azad Hossain, Head, ETE, CUET, who helped me to fulfill the requirement of my degree. I would like to specially thank Mohammad Mohiuddin, Alumnus, ETE, CUET, who, beside my thesis supervisor, spared his valuable time in helping me selecting a suitable topic with the merit to fulfill the requirement of my degree. I am grateful to my family that inspired me to pursue higher study in the first place. Their constant support and encouragement were the first thing that paved my way towards this journey. Last but not least, I would like to thank every official of the Department of Electronics and Telecommunication Engineering (ETE), CUET for helping me with technical and administrative supports during my thesis work.

Author

Abstract

Since brain tumors have a variety of forms, it is crucial to determine the types of tumors. The convolutional neural network (CNN) can extract features automatically with high-performance accuracy. The inability of this method is that it is unable to extract unknowing features both locally and globally, with a large number of parameters. As a result, it must deal with computational complexity and take a long time to train, additionally, due to a large number of parameters that occur overfitting. To address this, this research proposes an effective model to detect and classify brain tumors using a deep convolutional neural network. A novel Deep Parallel CNN (DPCNN) is used to extract the features locally and globally effectively. The proposed method also uses global average pooling technology instead of the fully connected layer which effectively reduces the number of parameters and solves the overfitting problem. In the initial stage, resized data is augmented to enhance the number of images in the pre-processing step. Next, the augmented data is applied to the DPCNN to extract the features both locally and globally in the feature extraction step. Lastly, different forms of classifiers like Softmax, KNN, and SVM are used to classify the various forms of tumors from MRI scans in the classification step. The proposed DPCNN-SVM architecture achieved a classification accuracy of 97.7% which is compared with other existing models.

বিমূর্ত

যেহেতু ব্রেন টিউমারের বিভিন্ন রূপ রয়েছে, তাই টিউমারের ধরন নির্ধারণ করা অত্যন্ত গুরুত্বপূর্ণ। Convolutional Neural Network (CNN)) উচ্চ-কর্মক্ষমতা নির্ভুলতার সাথে স্বয়ংক্রিয়ভাবে Features বের করতে পারে। এই পদ্ধতির অক্ষমতা হল যে, এটি প্রচুর পরিমাণে parameters Locally এবং Globally উভয়ই অজানা Features বের করতে অক্ষম। ফলস্বরূপ, এটিকে কম্পিউটেশনাল জটিলতার সাথে মোকাবিলা করতে হবে এবং প্রশিক্ষণের জন্য একটি দীর্ঘ সময় নিতে হবে, উপরন্তু, প্রচুর পরিমাণে Parameters এর কারণে ওভারফিটিং ঘটে। এটি মোকাবিলা করার জন্য, এই গবেষণাটি একটি Deep Parallel Convolutional Neural Network (DPCNN) ব্যবহার করে ব্রেন টিউমার detect এবং classify করার জন্য একটি কার্যকর মডেল প্রস্তাব করে। একটি নতুন DPCNN Locally এবং Globally কার্যকরভাবে বৈশিষ্ট্যগুলি বের করতে ব্যবহৃত হয়। প্রস্তাবিত পদ্ধতিটি সম্পূর্ণরূপে fully connected পরিবর্তে global average pooling প্রযুক্তি ব্যবহার করে যা কার্যকরভাবে প্যারামিটারের সংখ্যা হ্রাস করে এবং ওভারফিটিং সমস্যা সমাধান করে। প্রাথমিক পর্যায়ে, pre-processing ধাপে augmentation ব্যবহার করে resized ডেটা বর্ধিত করা হয়। এর পরে, Features extraction ধাপে Locally এবং Globally উভয় features extract করতে DPCNN -এ বর্ধিত ডেটা প্রয়োগ করা হয়। সবশেষে, Softmax, KNN, এবং SVM -এর মতো বিভিন্ন ধরনের Classifier ব্যবহার করা হয়। শ্রেণীবিভাগের ধাপে এমআরআই স্ক্যান থেকে বিভিন্ন ধরনের টিউমারকে শ্রেণীবদ্ধ করতে প্রস্তাবিত DPCNN-SVM আর্কিটেকচারটি ৯৭.৭% এর শ্রেণীবিভাগ নির্ভুলতা অর্জন করেছে যা অন্যান্য বিদ্যমান মডেলের সাথে তুলনা করা হয়।

Table of Contents

List of Figures

List of Tables

List of Abbreviations and Acronyms

AI	Artificial Intelligence
RBNN	Radial Basis Neural Network
DCNN	Deep Convolutional Neural Network
DL	Deep Learning
DT	Decision Tree
k-NN	K-Nearest Neighbors
SVM	Support Vector Machine
NF	Neuro-Fuzzy
MLP	Multi-Layer Perception
RBF	Radial Basis Function
PNN	Probabilistic Neural Network
RMS	Root Mean Square
FC	Fully Connected
BN	Batch Normalization
GAP	Global Average Pooling

Chapter 1: INTRODUCTION

This chapter explains the overview of the recent state of the world's machine learning in the brain tumor research field. Section 1.1 explains the motivation of the study using concise machine learning in the brain tumor field. That section 1.2 discusses the objective of the proposed system. The problem statement and thesis organization are described in sections 1.3 and 1.4, respectively.

1.1 Motivation

A brain tumor is a growth that occurs in the tissues surrounding the brain or skull and has a significant influence on human health. Figure 1.1 depicts the sample of brain tumor.

Figure 1.1: The sample of brain Tumor [1]

Different techniques can be used to distinguish between a benign and malignant development. Primary brain tumors, known as benign, arise in the brain, whereas secondary tumors, also known as brain metastases, are more usually caused by tumors outside the brain [1]. The brain tissue is strained by these tumors' irregular development. Pressure creates a number of cognitive problems that have a harmful impact on the body [2]. Patients with brain tumors may experience headaches, cognitive problems, difficulty walking or balancing, nausea and vomiting, muscular contractions, vision and hearing abnormalities, recession, dysphagia, mental disorders or confusion, and other symptoms [3].

There are many different types of brain tumors in the world, but the most well-known include glioma, meningioma, and pituitary tumors. Glioma tumors are found in glial cells with higher or lower grades, also known as astrocytes and cerebral hemispheres. The pituitary gland, on the other hand, is located in the front of the brain. Meningioma develops in the outer layer of the brain. Pituitary and meningiomas are benign, but gliomas are malignant [4]. Benign tumors can be healed with normal drugs or surgery, whereas malignant tumors can be controlled with radiotherapy and chemotherapy [5]. Figure 1.2 depicts the sample of brain tumor.

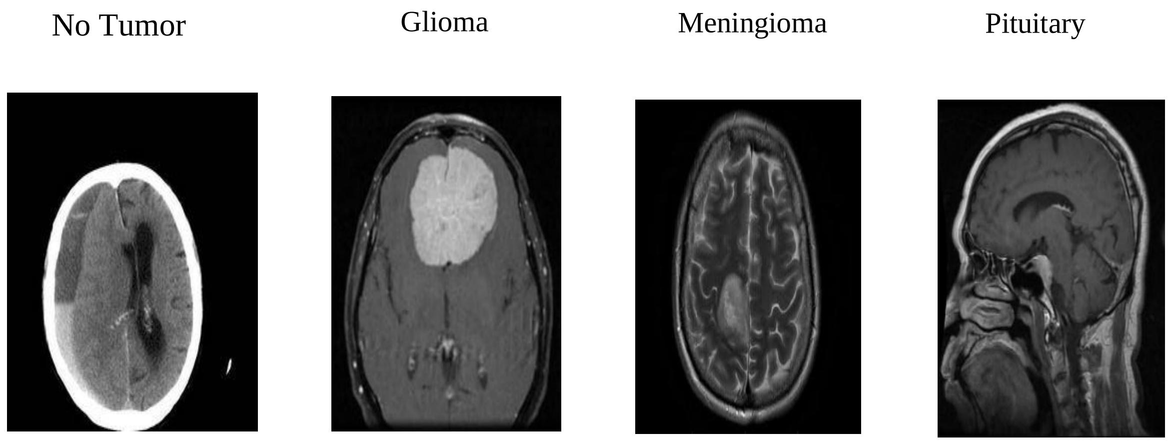


Figure 1.2: Original images of Kaggle Dataset [49]

The presence of brain tumors is significantly influenced by early diagnosis and classification. Numerous manual methods, including computed axial tomography (CT) scans, PET scans, and magnetic resonance imaging (MRI), can be used to diagnose it [6]. Although MRI is not often used to diagnose different kinds of malignancies, its interpretation is subject to human error and may be challenging to process large amounts of data. Figure 1.3 depicts the MRI image of a brain tumor.

Figure 1.3: The MRI images of brain Tumor [6]

Misclassification of brain tumors frequently results in the patient receiving the incorrect medication, which can be fatal. Brain surgery is carried out during a biopsy, another diagnostic procedure [7]. Many authors created automated techniques to identify malignancies from MRI scans ten years ago to avoid surgery and ensure error-free diagnosis. To identify and categorize different kinds of brain tumors from MRI, machine learning algorithms are being developed.

This research is tied to a few related papers. The authors Thejaswini et al. proposed a process for segmenting the input images using Adaptive Regularized Kernel-based Fuzzy C-means clustering (ARKFCM). Subsequently, a fusion of Artificial Neural Network (ANN) and Support Vector Machine (SVM) is employed to identify and categorize brain tumors with 91.4% accuracy, 98% sensitivity, and 78% specificity, respectively [8]. A hybrid intelligent technique for classifying MRI brain images was proposed by Ei-Dahshan et al. [9]. Using Discrete wavelet transform (DWT) for feature extraction, Principal Component Analysis (PCA) for feature reduction, and Feedforward Backpropagation Artificial Neural Network (Fp-ANN) and K-Nearest Neighbor (KNN) for feature classification in the end. The most significant drawback of this technique is that ANN cannot extract the characteristics automatically.

To address this issue, a decade ago, deep learning networks were established. The author G Hemanth used various techniques such as data collecting, pre-processing, average filtering, segmentation, feature extraction, and CNN via classification and identification [10]. A deep learning CNN technique using augmented data was described by Sajjad M. The VGG-19 CNN pre-trained model is set up for classification with an accuracy of 87.39% [11]. With 96.13 percent classification accuracy, Sultan implemented an architecture using CNN, a deep learning model [12]. A classification strategy using GLCM, a hybrid classifier, and stationary wavelet transform (SWT) was developed by Ginni Garg et al [13].

In order to modify the layers of GoogleNet, Reza presented the Deep Tumor Net model, which uses the Leaky Relu layer as an activation layer [14]. The Capsule network, a unique CNN architecture developed by Afshar et al., has a necessary accuracy of 90.89% [15]. Habib designed a convolutional neural network that leverages the Kaggle binary brain tumor classification dataset-I, which is used in this paper, to detect brain cancer. This method uses a modified neural network design to achieve an accuracy of 88.7% [16]. Although the pre-trained model trained the parameters over a large number of layers, resulting in a high accuracy rate, it requires significant computational time. Certain deep machine learning models, such as CNN, have the ability to automatically extract features from a lightweight design.

However, the disadvantage of this process is that it can arbitrarily extract unknown features, which can lead to overfitting issues.

The motivation for this study is to address the challenges stated, a novel Deep Parallel Convolutional Neural Network (DPCNN) for detecting and classifying the various kinds of brain tumors from MRI scans utilizing different classifiers.

1.2 Problem statement

Since brain tumors have a variety of forms, it is crucial to determine the types of tumors. The convolutional neural network (CNN) can extract features automatically with high-performance accuracy. The inability of this method is that it is unable to extract unknowing features both locally and globally, with a large number of parameters. As a result, it must deal with computational complexity and take a long time to train, additionally, due to the large number of parameters that occur overfitting.

1.3 Objectives

The main objective of this research is to develop an effective model system that can accurately detect and classify the brain tumor.

The objectives of this research are specified below-

i. To develop an effective pre-processing technique for brain tumor MRI images.
ii. To develop a novel Deep Parallel Convolutional Neural Network (DPCNN) topology to extract the features locally and globally from brain tumor MRI scans.
iii. To select an efficient classifier to classify brain tumors using obtained features.

iv. To perform a comparative analysis of the proposed system to study the improvements in efficiency and performance.

1.4 Thesis organization

The literature in this thesis is organized in five chapters. The contents of these chapters are briefly discussed as follows:

- Chapter 1, the introduction chapter, contains research motivation, objectives, and thesis organization. Motivation for the research, objectives for the study, and how this thesis paper is organized are concisely described in the chapter.
- Theoretical background in line with the literature review associated with the study is organized in Chapter 2. It will provide a brief overview of the theories,
- formulae, and literature required for the audience to grow interested in and understand the study.
- Chapter 3 systematically organizes the methodology framework followed in the study. A comprehensive flowchart for the methodological framework is illustrated for providing a concise understanding of how the study is performed. Details on designing and modeling various algorithms, along with design parameters and figures, are included in the chapter.
- Chapter 4 includes simulation results, performance analysis, and comparative study. Data and output figures are included in this chapter
- Finally, chapter 5 summarizes the overall study method, results, and findings. Challenges and limitations are also discussed. Future recommendations and conclusion finish the thesis paper.

Chapter 2: THEORETICAL BACKGROUND

This chapter contains a comprehensive theoretical background supported by a good amount of literature to help the reader understand. Section 2.1 explains the brain tumors, types of brain tumors, and brain tumor detection and classification. Section 2.2 describes the procedure of the pre-processing steps. Then, Section 2.3 describes the working principle of the convolutional neural network, including the Deep Parallel CNN. The pre-trained models like the GoogleNet, the VGG16, AlexNet, and ResNet50 are described in Section 2.4. Finally, Section 2.5 explains the working principles of classifiers, including K-Nearest Neighbour (K-NN), the softmax, Support Vector Machine (SVM), and Decision Tree (DT).

2.1 Brain tumors and types of brain tumor

A brain tumor is an uncontrollable development of abnormal tissues in the brain, often known as an intracranial tumor [17]. Brain tumors are separated into two types: primary and secondary. Primary brain tumors, also known as benign or non-cancerous, develop in the brain. Secondary brain tumors, also known as malignant or cancerous, develop in several regions of the body, including the stomach, breasts, lungs, and kidneys, before spreading to the brain [18]. Patients with brain tumors may experience headaches, cognitive problems, difficulty walking or balancing, nausea and vomiting, muscular contractions, vision and hearing abnormalities, recession, dysphagia, mental disorders or confusion, and other symptoms. There are many various types of brain tumors in the world, but the most well-known include glioma, meningioma, and pituitary tumors. Glioma tumors are found in glial cells with higher or lower grades, also known as astrocytes and cerebral hemispheres. The pituitary gland, on the other hand, is located in the front of the brain. Meningioma develops in the outer layer of the brain. Pituitary and meningiomas are benign, but gliomas are malignant [3-4].

2.1.1 Brain tumor detection and classification

Early diagnosis and classification of brain tumors have a significant impact on patient fatality rates. It can be diagnosed by a variety of manual approaches, including magnetic resonance imaging (MRI), positron emission tomography (PET), and computed axial tomography (CT) scans [6]. MRI is infrequently used to diagnose many types of malignancies, but it is subject to human interpretation and can be difficult to analyze with a large amount of data. Misclassification of brain tumors can frequently result in the patient receiving the wrong medication, which is life-threatening. Another diagnostic approach is a biopsy, which involves brain surgery [7]. To prevent surgery and error-free diagnosis a decade ago, many authors devised automated algorithms to spot cancers from MRI data. Machine learning algorithms are being developed to detect and categorize various forms of brain tumors using MRI. Using Discrete wavelet transform (DWT) to extract features, Principal Component Analysis (PCA) to reduce features, and Feedforward Backpropagation Artificial Neural Network (Fp-ANN) and K-Nearest Neighbor (KNN) to classify features, Ei-Dahshan et al. proposed hybrid intelligent techniques for MRI brain image classification [9]. Many efforts have been undertaken to develop an extremely reliable and efficient system for automatically categorizing brain tumors. Traditional machine learning approaches rely on handmade qualities, which increases the cost and limits the longevity of the solution. However, in some cases, supervised learning models outperform unsupervised learning techniques, resulting in an overfitted model that is unsuitable for use with another large database. These issues highlight how important it is to develop a fully automated, machine learning-based system for brain tumor classification. CNN's architecture is based on a deep learning model, a neural network that is particularly good at classifying and recognizing images. [19,20].

2.2 Pre-processing

In the medical imaging field, preprocessing of the image is a most exciting and challenging task to provide accurate details for the diagnosis process, such as tumor identification & noise removal. Therefore, it needs an effective data preprocessing process. There are various types of pre-processing steps are used in machine learning like image resize, gray scale conversion, augmentation etc.

2.2.1 Image resize

In machine learning, image resizing is crucial for pre-processing images to ensure consistency in input size across a model. These are a few popular methods for resizing images are: Nearest Neighbor Interpolation, Bilinear Interpolation, Lanczos Resampling, Padding and Cropping, Random Resizing, Aspect Ratio Preservation etc.

2.2.2 Gray scale conversion

Grayscale image conversion is an essential preprocessing step in machine learning and image processing. Because grayscale images only have one channel rather than three (RGB), image processing tasks can be made simpler and computational complexity can be decreased. Here a few Gray Scale conversion methods are: Averaging Method (Luminosity Method), Luminance Channel Extraction, Principal Component Analysis (PCA), Single Channel Conversion, Using Libraries and Tools etc.

2.2.3 Augmentation

Image augmentation is an essential machine learning technique that applies several changes to pre-existing pictures to enhance the size and diversity of a dataset. The following lists two categories of image augmentation:

In computer vision and machine learning, color space augmentation is a technique that modifies the color characteristics of images to increase the performance and robustness of models. This technique entails altering the colors of an image in ways that can help a model generalize better to changes in lighting, color, or other environmental factors. Some common methods of color space augmentation: Color Jittering, Color Space Conversion, Histogram Equalization, Channel Shuffling, Color Noise, Random Color Transformations.

The spatial characteristics of objects or images can be altered through the application of geometric transformations. This technique is applied to enhance data, increase robustness, or simulate different viewpoints. Some common geometric transformations are: Translation, Rotation, Scaling, Shearing, Affine Transformation, Flipping, Cropping, Zooming etc.

2.3 Convolutional neural networks (CNNs)

Neural networks are the foundation of deep learning techniques, which are a subset of machine learning. They consist of node layers with one or more hidden levels, an input layer, and an output layer. Each node has a weight and threshold that are related to each other. If the output of a node exceeds a predetermined threshold value, the node is triggered, and data is sent to the network's subsequent layer. If not, no data is passed to the next tier of the network. Although the most of that article was committed to feedforward networks, different types of neural nets are utilized for a variety of purposes and data types. For example, convolutional neural networks, or CNNs or ConvNets, are more commonly utilized in computer vision and classification applications than recurrent neural networks, which are more commonly used in speech recognition and natural language processing applications. Before CNNs were developed, object identification in images needed time-consuming, manual feature extraction approaches. Conversely, convolutional neural networks now provide a more scalable way to identify objects in images and classify images by employing the ideas of linear algebra and matrix multiplication to identify patterns in images [24,25,26]. The most basic CNN model consists of one trainable feature extraction step and one classification step [27]. The feature extraction step has three layers: a convolutional layer, an activation layer, and a pooling layer [28]. The classification step consists of multiple fully connected layers of a multilayer perceptron. Figure 2.1 depicts the basic structure of CNN [29].

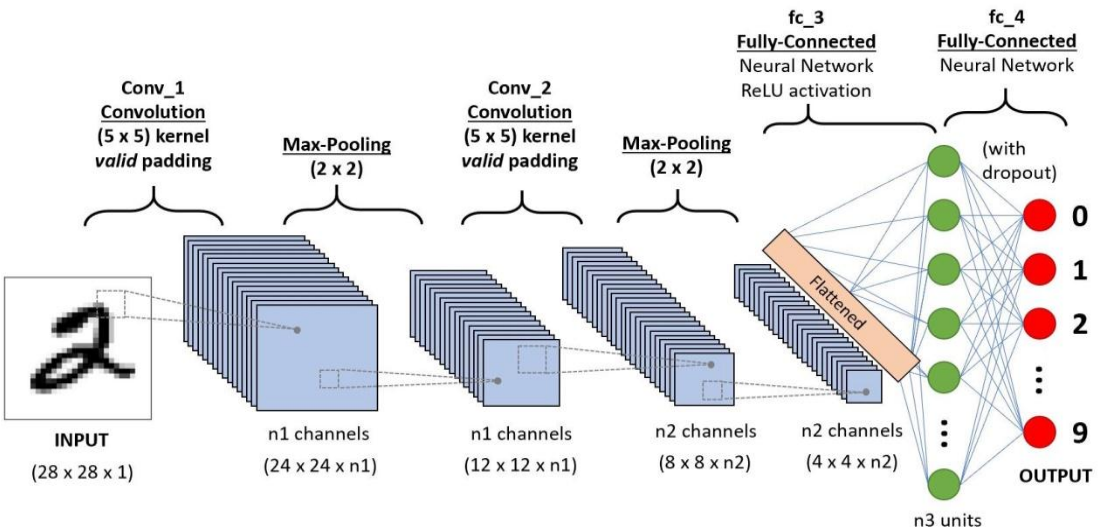


Figure 2.1: The basic structure of the CNN [29]

2.3.1 Convolution Layer

The convolutional layer is the essential part of a CNN. The parameters of the set of filters, also known as kernels, that it contains must be learned throughout the training process. Generally speaking, the filters are not as big as the image. An activation map is produced by each filter convolving with the image. Convolution involves sliding the filter across the height and width of the image, and computing the dot product between each filter element and the input at each spatial position [30]. Figure 2.2 depicts the calculation technique's main schematic illustration of a convolution operation.

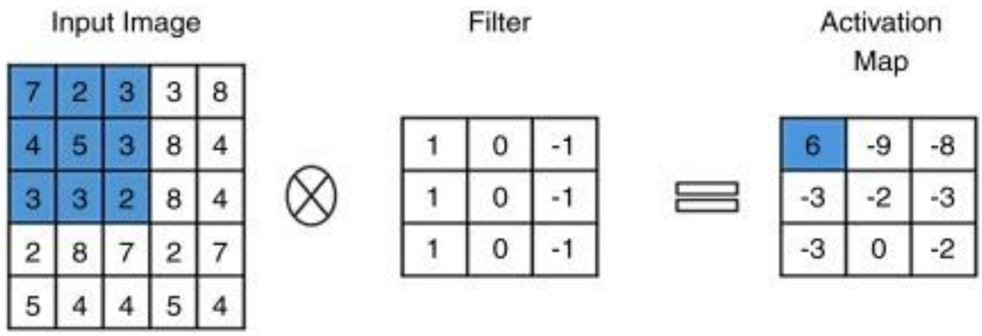


Figure 2.2: Visualization of the convolution operation [30]

Figure 2.2 shows how the convolution kernel can be shifted along with the up-down and left-right directions in accordance with the stride on the input feature map. The rectangular convolutional kernel can be used in the feedforward calculation process to cross every element on the whole input feature map in order to obtain the output feature maps of the convolutional layer. A multichannel input feature map's convolution process is as follows [31]:

$$\boldsymbol{X_i^k} = \sum_{\boldsymbol{C=1}}^{\boldsymbol{C}} \boldsymbol{W_i^{(c,i)}} * \boldsymbol{X_{i-1}^C} + \boldsymbol{B_i^k} \qquad (2.1)$$

Where $*$ is the convolutional operator; i is the network layer index; k is the convolution layer's index number; the kth is the group convolution kernel; C is the channel index number; the input feature map of channel c is $\boldsymbol{X_{i-1}^{C}}$; The ith layer's convolution kernel's weight is $\boldsymbol{W_i^{(c,i)}}$; $\boldsymbol{X_i^k}$ is the kth output feature map obtained following convolution of the kth group convolution kernel with the input feature map; The bias of the ith layer's kth group filter is $\boldsymbol{B_i^k}$.

2.3.2 Activation functions

In machine learning, an activation function is a mathematical operation that takes advantage of the model's non-linearity. Every neuron's output is where it functions. It generates the output by gripping the weighted sum from the first layer and then moves on to the next layer. Activation functions come in a variety of forms, including Clipped ReLU (CReLU), Leaky ReLU (LReLU), and Rectifier Linear Unit (ReLU).

The function of activation to improve performance and lessen gradient diffusion, ReLU is frequently used after the convolutional layer [33]. The equation for the ReLU function is as follows:

$$f(g) = \max(0, g) \qquad (2.2)$$

When the value of g is 0 or negative in this equation, it denotes that the output is also zero. When g's value is greater than zero, it equals 1. However, this function's disadvantage is that it only accepts positive input; if it receives a negative number, the learning process will be terminated.

The output of Clipped ReLU is presented as a threshold operator with upper and lower thresholds. The clipped ReLU equation is;

$$Clipped\ ReLU\ (x) = \pi \max(a, \min(x, b)) \qquad (2.3)$$

The smoother training introduced by Clipped ReLU is an issue. The performance of learning is impacted by this function, which needs to be adjusted but cannot be done precisely.

Leaky Relu (LReLU) is a refined version of Clipped ReLU and ReLU that can address the problems and deliver optimal results. The formula for Leaky ReLU is;

$$F(x) = \max(\alpha x, x) \tag{2.4}$$

2.3.3 Pooling Layer

The convolution neural network's feature map's spatial dimensions are decreased and downsampled through the use of a pooling layer. Because a deep neural network has a large number of feature maps, overfitting and computational issues arise. To solve this issue, a pooling layer operation is carried out. Max pooling and Average pooling are the two pooling techniques used by CNN the most frequently.

Max pooling considers the subset's maximum value. Max pooling ignores some relevant information because it considers only the strongest feature elements.

Average pooling is utilized to solve this issue. By using average pooling, the subset's average value is obtained. Even if every element is chosen by average pooling, each element is always given equal weight and contributes all necessary elements. Compared to Max pooling, it extracts features more smoothly [32]. Figure 2.3 shows the calculating process, principle schematic diagram, and pooling operation.

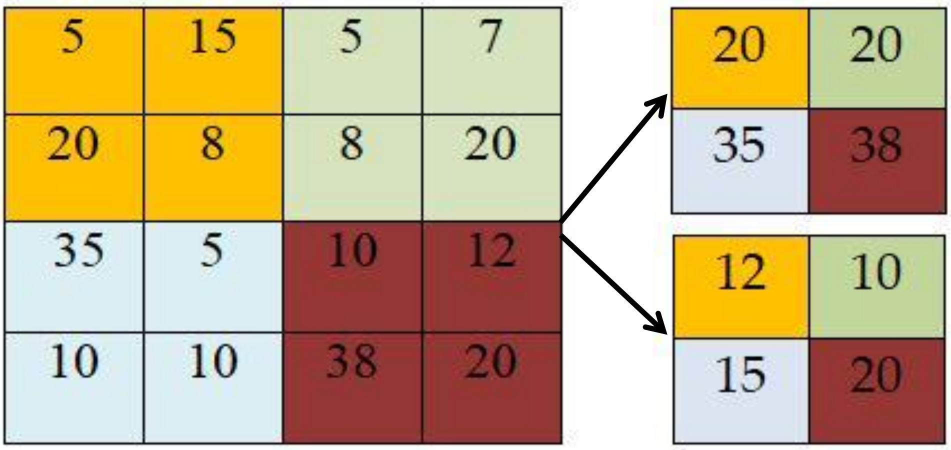


Figure 2.3: Pooling operation

2.3.4 Fully Connection Layer

Traditional CNN architecture employs a fully connected layer during the classification step. A fully connected layer in a traditional CNN typically consists of two to three layers that are fully connected to a feedforward neural network [33]. A CNN's flattening function converts the final output feature map into a one-dimensional array. The main responsibility is to extract features from CNN's output data and connect the Softmax classifier to the feature extraction stages. All neurons between layers are interconnected in a network that is fully connected, as described below:

$$\boldsymbol{o}(\boldsymbol{X}) = \int (\boldsymbol{W} \times \boldsymbol{X} + \boldsymbol{B}) \tag{2.5}$$

Where the activation function is denoted by f (x), A fully connected layer's input is denoted by X, W, and B are the weights and biases of a fully connected network and the result of a fully connected layer is $o(X)$. Visualization of the fully connected layer is shown in Figure 2.4.

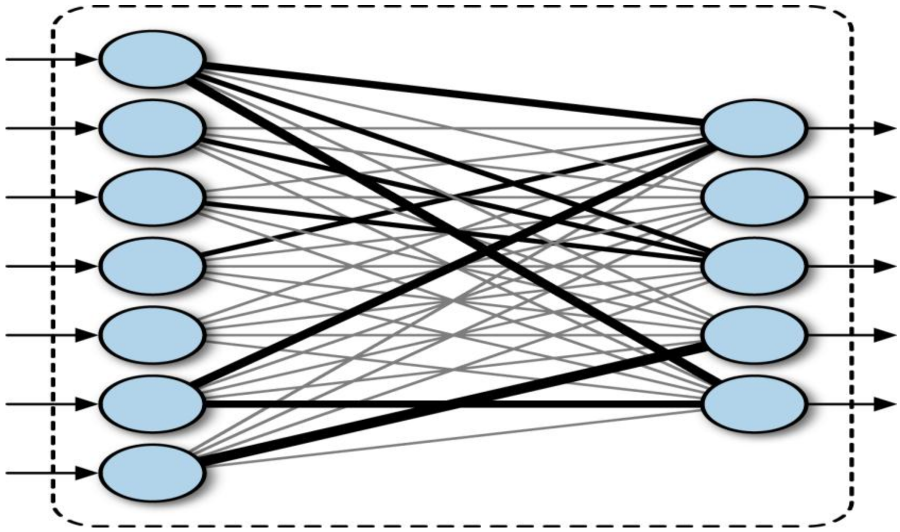


Figure 2.4: Visualization of the fully connected layer [33]

2.3.5 Parallel convolutional neural network (DPCNN)

When multiple CNN models or branches perform simultaneously, the architecture is referred recognized as a parallel convolutional neural network, or parallel CNN. Parallel CNNs may have the same or different number of layers. The CNN outputs serve as inputs for a fully connected back propagation neural network (BPNN). The suggested model's structure can reduce CNN complexity by reducing the total number of CNN layers. Better representations for tasks like classification, segmentation, or object recognition are made possible by these networks' ability to analyze various input types or feature levels simultaneously or to extract multiple feature levels simultaneously. By combining multiple paths, the model can capture a richer set of features and also improving feature extraction efficiency [34]. Basic architecture of the parallel is shown in Figure 2.5

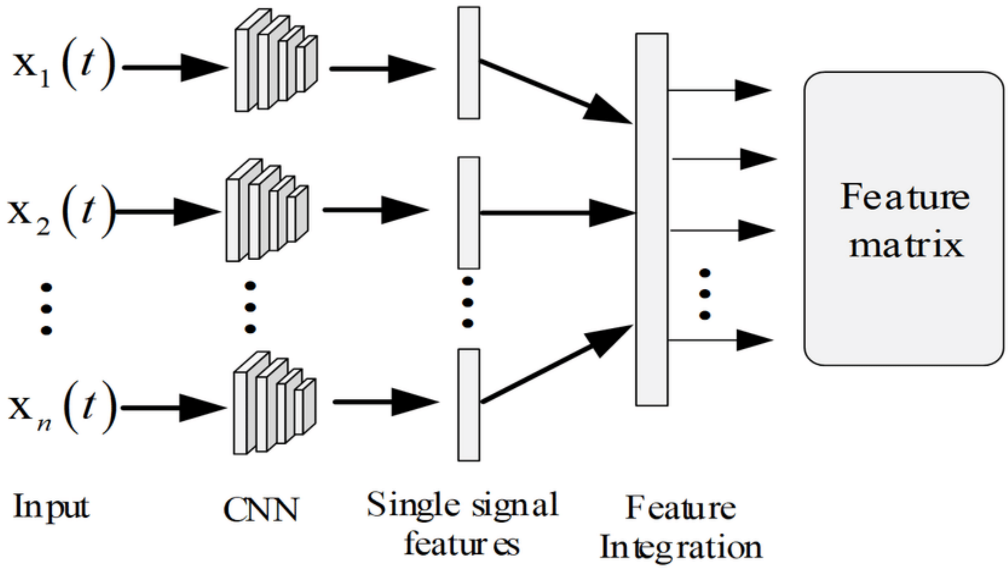


Figure 2.5: Basic architecture of the parallel CNN [34]

2.4 Pre-trained models

A pre-trained model is one that was constructed by someone else and trained on a large dataset to address a related issue. Many deep learning pre-trained models, like GoogleNet, VGG16, AlexNet, and ResNet50, are built on ImageNet. These models are capable of extracting high-level features from images and can be utilized for image classification, object detection, face recognition, and other applications [35]. There are various types of pre-trained models used in deep learning, like GoogleNet, VGG16, VGG19, ResNet18, ResNet50, etc.

2.4.1 GoogleNet

Google Net was proposed by Google researchers in their 2014 research paper "Going Deeper with Convolutions" [36]. The GoogLeNet architecture differs significantly from earlier state-of-the-art designs such as AlexNet. It employs a number of techniques, such as global average pooling and 1×1 convolution, to produce a deeper architecture [37]. We offer three initiation structures that might be used in different scenarios. Before employing the costlier 3×3 and 5×5 convolutions, Inception typically computes reductions using a 1×1 convolution. It functions as an example of how to design an architecture with high capacity. Rather than relying just on more powerful technology, bigger datasets, and larger models, most identifying capabilities advances rely on new ideas, algorithms, and developed network designs [37]. Figure 2.6 illustrates the design of GoogleNet.

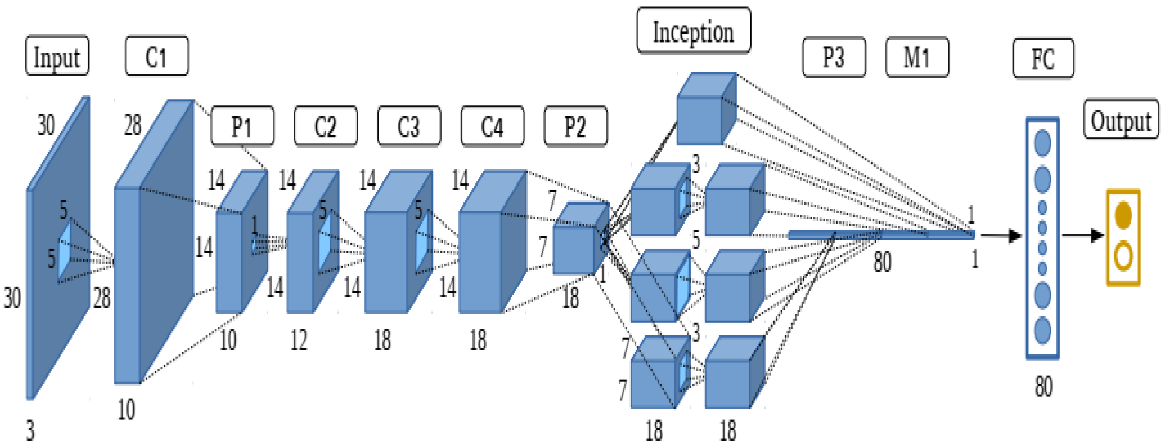


Figure 2.6: Architecture of the GoogleNet [37]

2.4.2 VGG16

Andrew Zisserman and Karen Simonyan created the deep CNN known as VGG. Thirteen convolutional layers, three fully linked layers, and five max-pooling layers make up VGG16. Consequently, there are sixteen levels in all with programmable settings. This explains the model's VGG16 name. This architecture's growing network structure is its most important component. [38]. The VGG-16 has a 224 by 224 px input layer and a 3 x 3 filter. The internal structure of this structure consists of three completely linked layers, a pooling layer, and five convolutional layers. The last layer in classification processes is called the softmax. The VGG16 architecture is detailed in Figure 2.7

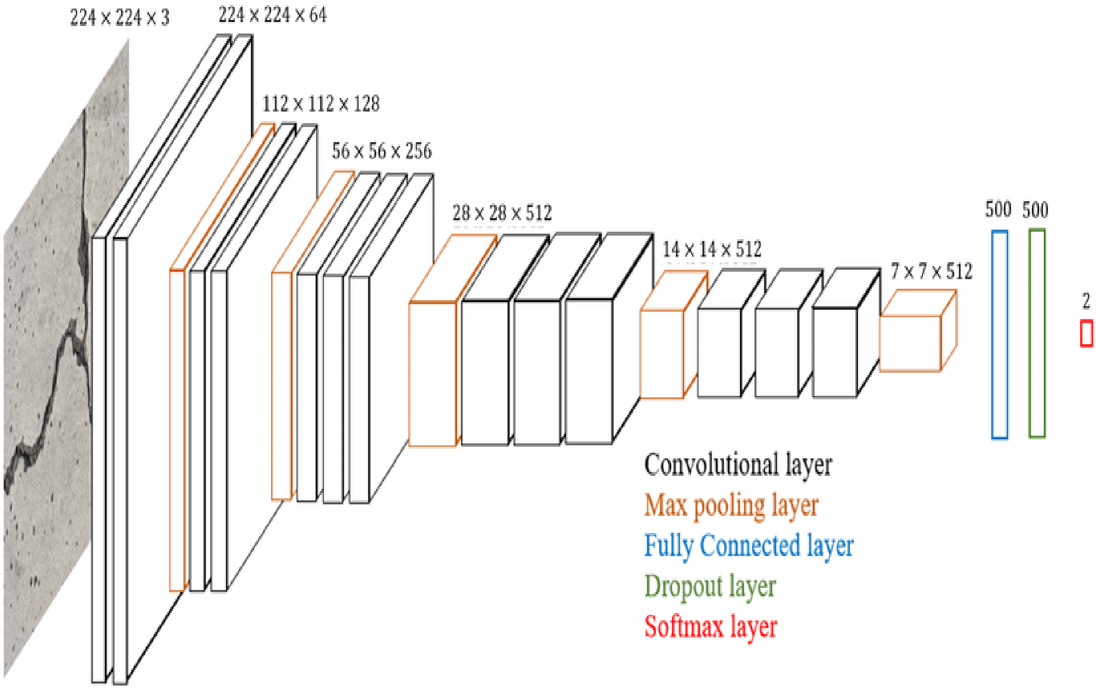


Figure 2.7: Architecture of the VGG16 [38]

2.4.3 VGG19

The architecture of the VGG19 network is comparable to that of the AlexNet, with increasing filters as you move further into the network and successive convolutional layers. The model has five pooling layers with the maximum pooling approach with 2 × 2 windows, three fully connected layers, and sixteen convolutional layers. Since the perceptual field was found to be just as efficient as larger filters, the architecture promoted the use of smaller filters. Smaller filter sizes also result in fewer training parameters. [39]. Figure 2.8 shows the VGG19 architecture in detail.

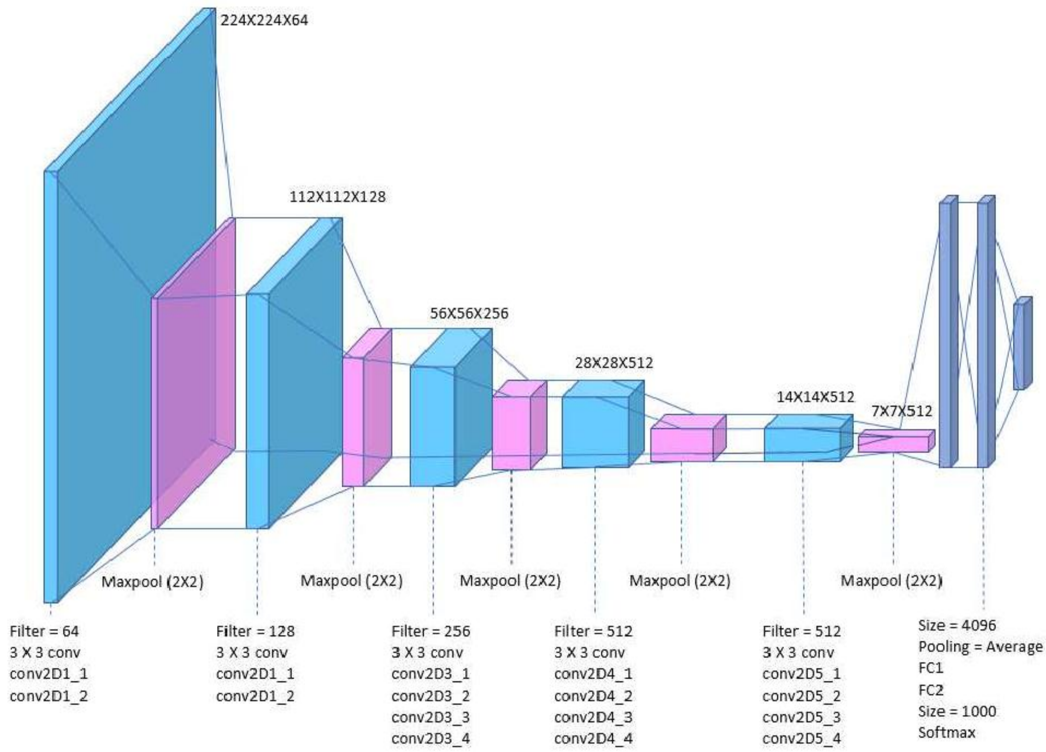


Figure 2.8: Architecture of the VGG19 [39]

2.4.4 ResNet18

ResNet18 has eighteen deep layers and a 72-layer design. This network's architecture aims to facilitate the effective operation of many convolutional layers. About 11 million parameters may be trained on Resnet18. It is made up of CONV layers with 3x3 filters, much like VGGNet. Throughout the network, just two pooling layers are utilized: one at the start and one at the end. There are identity connections between each pair of CONV levels [40]. Architecture of the ResNet18 is shown in Figure 2.9

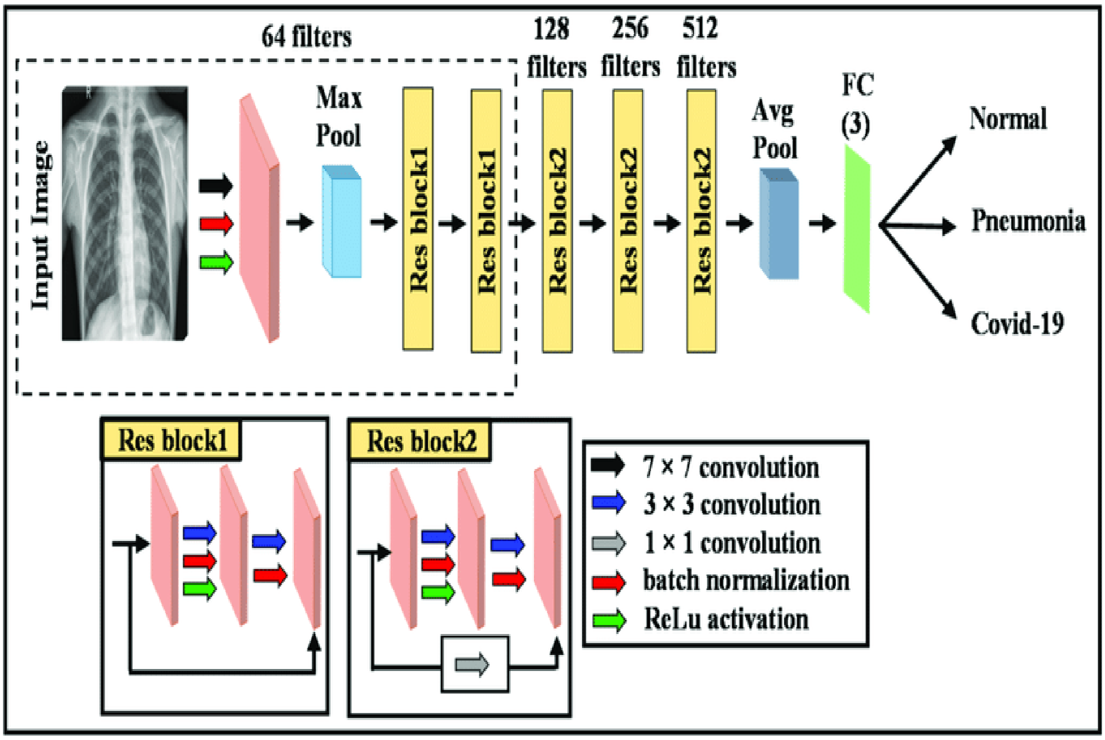


Figure 2.9: Architecture of the ResNet18 [40]

2.4.5 ResNet50

CNNs with more than 50 layers, resembling the well-known ResNet-50 model, are called deep residual networks. ResNets' main concept is to use shortcut connections to bypass convolutional layer blocks [41]. Convolutional layers with a stride of two directly execute down sampling, and batch normalization happens right after each convolution and just prior to ReLU activation. The identity shortcut is used when the dimensions of the input and output are the same. 1×1 convolution is used as a projection shortcut to match larger dimensions. The shortcuts employ a stride of two when they cross feature maps of different sizes. The network's 1,000 fully connected (fc) layer, which is enabled via softmax, is its pinnacle. A total of 50 weighted layers with 23,534,592 trainable parameters are present [44]. The architecture of the ResNet-50 is shown in Figure 2.10.

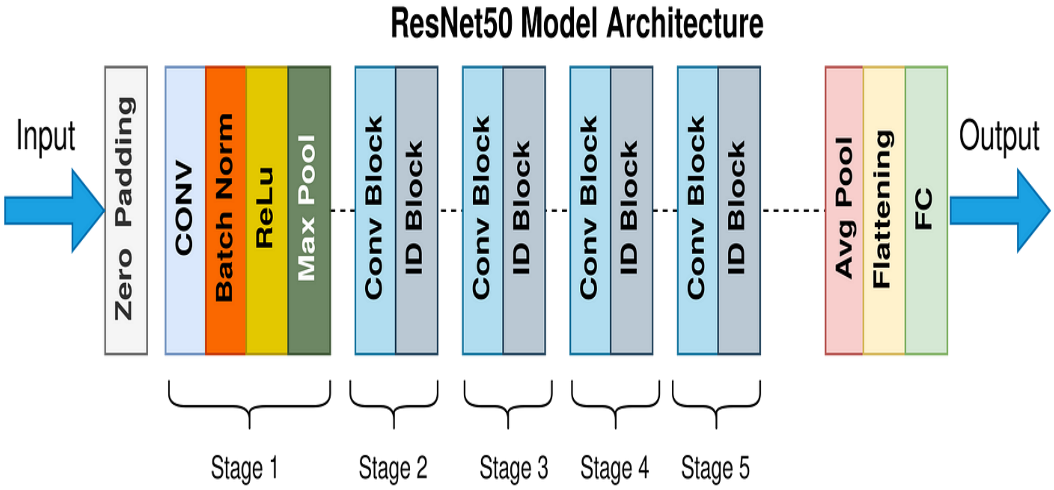


Figure 2.10: Architecture of the ResNet50 [44]

2.5 Classifiers

In data science, a classifier is a type of machine learning algorithm that classifies data inputs. Labeled data is used to train classifier algorithms; for example, in the case of image recognition, the classifier receives training data that includes image labels [45]. When the classifier has had enough training, it may accept unlabeled photos and generate categorization labels for each one. Classifier algorithms utilize advanced mathematical and statistical techniques to predict the likelihood that a data input will be classified in a specific way. Deep learning uses a variety of classifiers, including K-Nearest Neighbor (K-NN), Softmax, Support Vector Machine (SVM), Decision Tree (DT), and so on.

2.5.1 Softmax Classifier

The softmax classifier is a supervised learning method that is commonly used when there are multiple classes involved, as opposed to the logistic regression classifier, which is used for binary class classification. Each class in the Softmax classifier is assigned a probability distribution [46]. All other probabilities are scaled using the probability distribution of the class with the highest likelihood, which is normalized to one. The output of neurons is also turned into a probability distribution over the classes using a softmax function. Softmax is the mathematical operation that is used at the output layer to transform extracted features into a probability distribution. The formula of the softmax equation is given below

$$softmax\ (z)_i = \frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_j}} \tag{2.6}$$

In this equation, Z represents the input vector, e is the natural logarithm base, n is the number of classes in the multi-class classifier, e^{z_i}) represents the standard exponential scores of the input, and $\sum_{j=1}^{n} e^{z_j}$ represents the sum of the standard exponential scores of the output.

2.5.2 KNN Classifier

K-Nearest Neighbors (KNN) is a supervised approach used for classification and regression problems. KNN is used to determine the proximity of neighbors. Assign the most frequent class among the neighbors to the new data point for classification, and use the average value or weighted average in regression [47]. Because of its simplicity and efficiency, K-Nearest Neighbor is a popular classifier. It has been referred to as a lazy learner because its training phase consists just of storing all training examples as classifiers and it waits to decide how to generalize beyond the training data until it encounters each new query instance. The K-NN algorithm classifies a new data point based on how similar the stored and current data are. This indicates that the K-NN algorithm can quickly classify fresh data when it appears in a suitable category.

2.5.3 SVM Classifier

The Support Vector Machine (SVM), one of the most popular supervised learning algorithms, can be used to resolve problems with both regression and classification. The goal of the SVM algorithm is to determine the optimal line or decision boundary that can partition n-dimensional space into classes so that we can categorize new data points in the future very quickly [47]. A hyperplane is the name of this best decision boundary. In order to create the hyperplane, SVM selects the extreme points and vectors. The term "support vectors" refers to these extreme cases, and it is the name of the SVM algorithm. Finding the decision boundary or hyperplane that can best divide data points into distinct classes is the central idea of the Support Vector Machine (SVM). In order for it to quickly classify the new data point in the appropriate category in the future [48]. Figure 2.11 shows the architecture of the SVM classifier. The formula of the SVM equation is given below;

$$W.X + B = 0 \qquad (2.7)$$

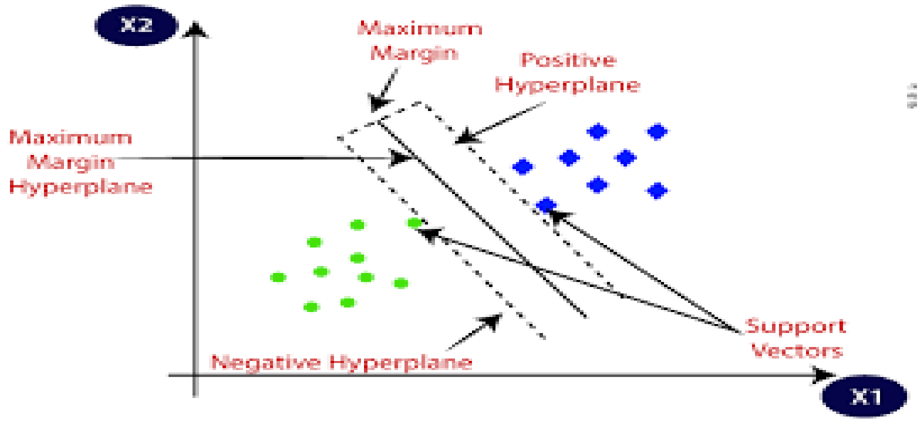


Figure 2.11: Architecture of the SVM [48]

A hyperplane's equation is w. x + b =0, where b is the offset and w are the vector normal to the hyperplane. We can declare a point to be positive if the value of w. x + b > 0; otherwise, it is a negative point.

Chapter 3: METHODOLOGY AND MATERIALS

This chapter discusses detail methodology and materials used in the research. Section 3.1 explains the methodology. Section 3.2 describes the experimental data pre-processing techniques. Then section 3.3 explains the feature extraction procedure including. The best classifier selection techniques are described in section 3.4.

3.1 Methodology

Due to the disadvantage of deep CNN, it can arbitrarily extract unknown features, which can lead to overfitting issues. To address the challenges stated, a novel DPCNN for detecting and classifying the various kinds of brain tumors from MRI scans utilizing different classifiers. Figure 3.1 shows this proposed method, which consists of three essential phases, for determining and classifying the different kinds of brain tumors using MRI data. The three steps are: (1) pre-processing the data; (2) feature extraction; and (3) classification. Data are downsized without deleting any content during the pre-processing stage. Then, grayscale conversion is carried out, and lastly, data are augmented in pre-processing step. The feature extraction process, which comes in second, involves choosing hyperparameters and creating a DPCNN to extract features both locally and globally. The third and last phase is the classifier step, in which several classifiers, including Softmax, KNN, and SVM, are used to classify distinct kinds of brain tumors.

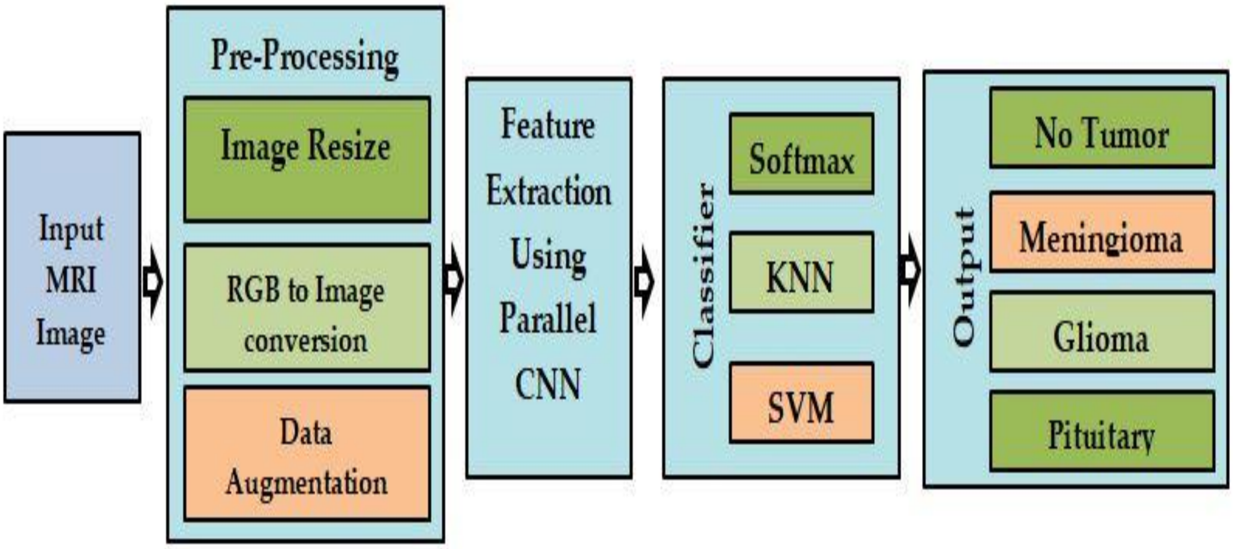


Figure 3.1: Flowchart of the proposed system

3.2 Data pre-processing

To reduce unwanted distortions and improve the performance some pre-processing steps are done before training the suggested model. The pre-processing step contains three steps first is image resizing, second is grayscale conversion and third is data augmentation accordingly.

3.2.1 Image Resize

A dataset is a collection of images used to train or analyze a model. This model uses datasets from the free source "Kaggle" [49]. The MRI pictures in this collection include four categories of brain images: glioma (826 images), meningioma (822 images), no tumor (395 images), and pituitary (827 images). The original images of the Kaggle dataset are 256 × 256. Figure 3.2 depicts the four types of brain tumor photos in the Kaggle dataset. As a result, 80% of the data is loaded for training, with 20% loaded for testing.

No Tumor Glioma Meningioma Pituitary

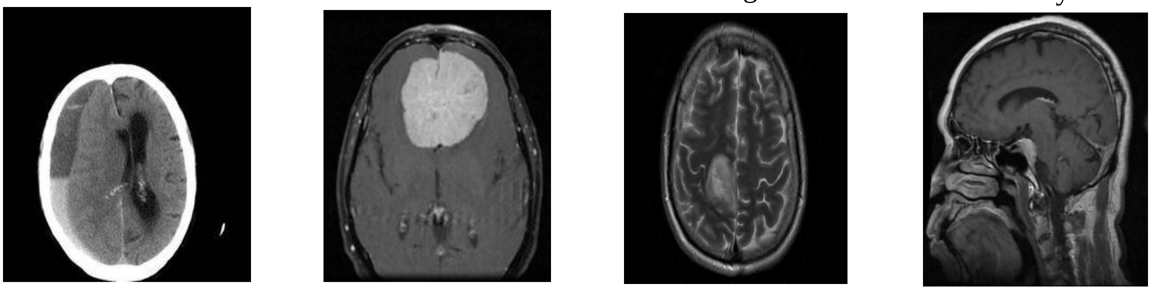
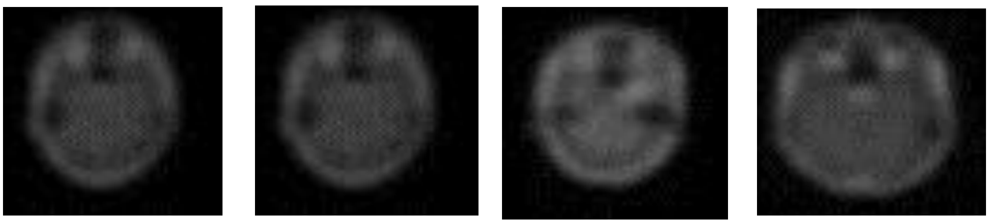


Figure 3.2: Original image of the Kaggle dataset [49]

Before training the proposed technique, resizing images in the dataset is a common pre-processing step. Since the original images have different dimensions, scaling an image is necessary to enhance efficiency and remove computational constraints. The dataset's photos are downsized from (256×256) to (60×60) dimensions without losing any content in order to produce a simple calculation.

MATLAB offers a variety of functions and methods for scaling images. These can be combined with various interpolation techniques and tactics to accommodate certain use cases, such as object recognition, segmentation, and image classification.



No Tumor Glioma Meningioma Pituitary

Figure 3.3: Resized images

3.2.2 RGB to grayscale image conversion

Grayscale images have one dimension, but raw photos from datasets with three colors have three dimensions. RGB images are converted to grayscale images.

The simplest way to convert a picture to grayscale is to take the RGB (red, green, and blue) pixel values and average them. Nevertheless, a weighted average is utilized since the human eye interprets brightness differently across the color spectrum (perceiving green as brighter than red or blue).

Formula:

$$Gray=0.2989\times R+0.5870\times G+0.1140\times B \qquad (3.1)$$

Since the green channel is more observably pleasing to the eye, this formula gives it more weight in consideration of the human visual system.

3.2.3 Data augmentation

The last phase of the pre-processing phase is data augmentation. There aren't enough images for testing and training, which could lead to an overfitting issue. To increase the number of images in the dataset before loading it for training, data augmentation is used to reduce misclassification and overfitting issues. Different geometric modifications, such as scaling, shearing, and translating, are used to enhance the photographs. The following describes how to set the parameter for the recommended augmenting operator:

- Translation-translate with the range [-3 3] along the X-axis and [-3 3] along the Y-axis respectively.
- Shear-Shear with the range [-0.05 0.05] along the X-axis and [-0.04 0.04] along the Y-axis respectively.
- Scale-Scale with the value [1 2] along horizontal and vertical.

3.3 Feature extraction step

Feature extraction is the process of obtaining features from data. This stage consists of two parts: the DPCNN section and the hyper-parameter selection part.

3.3.1 Proposed DPCNN

These days, deep learning networks are greatly influenced by CNN. The CNN performs admirably when it comes to autonomously extracting features from MRI scans and classifying them. CNN's inability to randomly extract unknown local and global properties is one of its notable drawbacks. The present research proposes DPCNN with Two Pathway CNN to detect and categorize brain tumor images by extracting known features both locally and globally. Two pathways CNN contain local pathway and global pathway within 26 layers. The local pathway contains 2D convolutional layers with a kernel size of (3×3,5×5,7×7) and the global pathway contains 2D convolution layers with a kernel size of (12×12,15×15,17×17) accordingly, the stride size of this architecture is (1,1) and same padding is used.

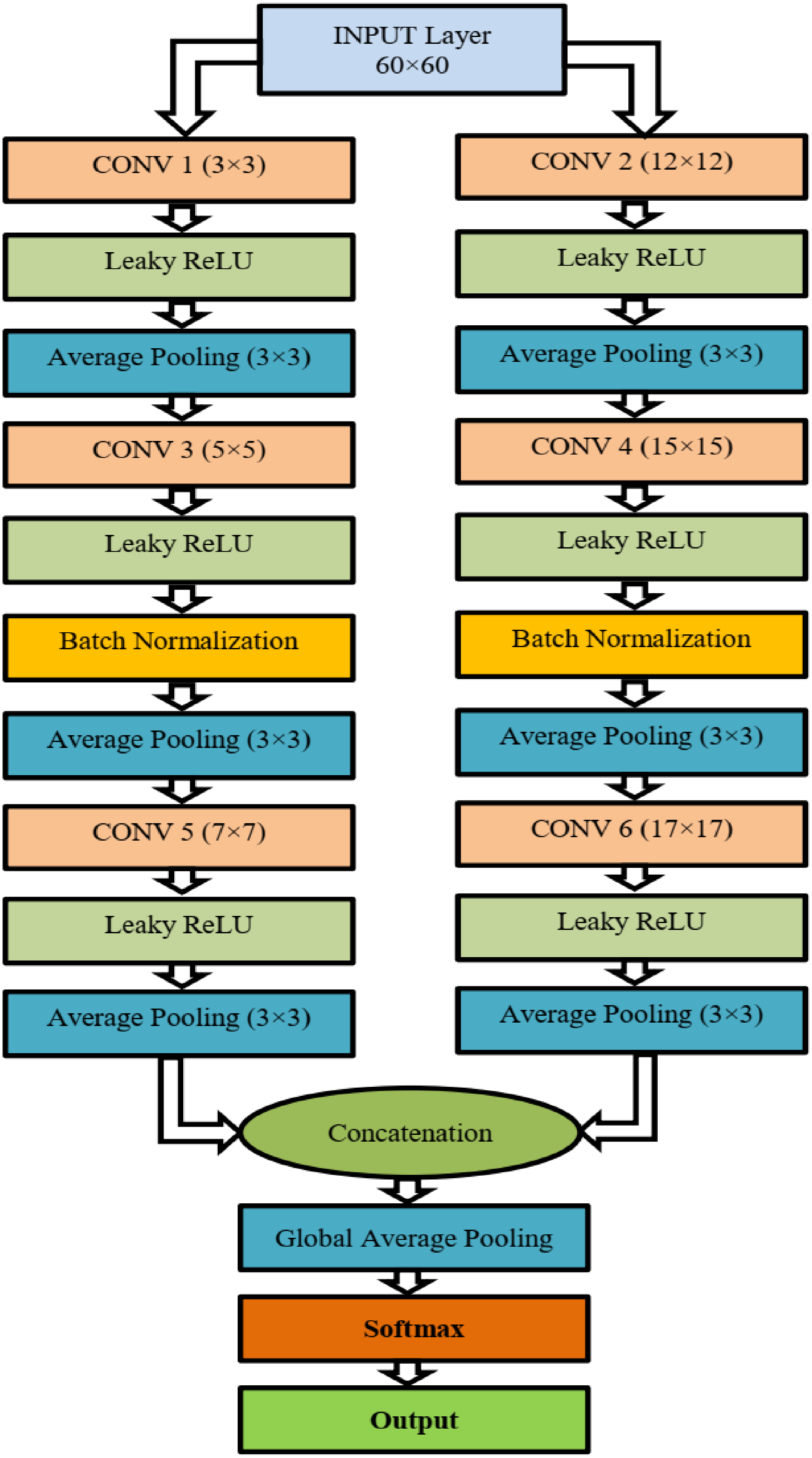


Figure 3.4: Proposed deep parallel CNN model

Although the Batch normalization (BN) and the activation function Leaky ReLU (LReLU) layer is used to remove overfitting. Downsampling the dimension of the output feature map average pooling layer is used with the kernel size (3×3). Both pathways are connected by a concatenation layer. Global average pooling is used instead of a fully connected layer. Various types of classifiers such as Softmax, KNN, and SVM are used to classify the brain tumor images and finally, DPCNN-SVM provides the best accuracy. Proposed deep parallel CNN model is shown in Figure 3.4

Table 3.1: Hyper-parameter of Deep Parallel CNN

Sl No.	Name	Type	Activations	Learnable
01	60×60×1 Images With" Zerocenter Normalization	Image Input	60×60×1	-
02	Conv _2 64 12×12×1 Convolutions With Stride [1 1] and Padding 'Same'	Convolution	60×60×64	Weights 12×12×1×64 Bias 1×1×64
03	Conv _1 64 3×3×1 Convolutions With Stride [1 1] and Padding 'Same'	Convolution	60×60×64	Weights 3×3×1×64 Bias 1×1×64
04	Leakyrelu _1 Leaky ReLU with a scale 0.01	LReLU	60×60×64	_
05	avgpool2d _1 3×3 average pooling with stride [1 1] and padding 'same'	Average pooling	60×60×64	_
06	Leakyrelu _2 Leaky ReLU with a scale 0.01	LReLU	60×60×64	_
07	avgpool2d _2 3×3 average pooling with stride [1 1] and padding 'same'	Average pooling	60×60×64	_

08	Conv_4 32 15×15×64 convolutions with stride [1 1] and padding 'same'	Convolution	60×60×32	Weights 15×15×64×32 Bias 1×1×32
09	Leakyrelu _4 Leaky ReLU with a scale 0.01	LReLU	60×60×32	_
10	Batchnorm _ 2 Batch normalization with 32 channels	BN	60×60×32	0ffset 1×1×32 Scale 1×1×32
11	Avgpool2d _ 4 3X3 average pooling with stride [1 1] and padding 'same'	Average pooling	60×60×32	_
12	Conv_6 32 17×17×32 convolutions with stride [1 1] and padding 'same'	Convolution	60×60×32	Weights 17×17×32×32 Bias 1×1×32
13	Leakyrelu _6 Leaky ReLU With a scale 0.01	LReLU	60×60×32	_
14	Avgpool2d _ 6 3X3 average pooling with stride [1 1] and padding 'same'	Average pooling	60×60×32	_
15	Conv_3 32 5×5×64 convolutions with stride [1 1] and padding 'same'	Convolution	60×60×32	Weights 5×5×64×32 Bias 1×1×32
16	Leakyrelu _3 Leaky ReLU with a scale 0.01	LReLU	60×60×32	_
17	Batchnorm_1 Batch normalization with 32 channels	BN	60×60×32	Offset 1×1×32 Scale 1×1×32
18	Avgpool2d _ 3 3×3 average pooling with stride [1 1] and padding 'same'	Average pooling	60×60×32	_

19	Conv_5 32 7×7×32 convolutions with stride [1 1] and padding 'same'	Convolution	60×60×32	Weights 7×7×32×32 Bias 1×1×32
20	Leakyrelu _ 5 Leaky ReLU with a scale of 0.01	LReLU	60×60×32	Offset 1×1×32 Scale 1×1×32
21	Avgpool2d _5 3×3 average pooling with stride [1 1] and padding 'same'	Average pooling	60×60×32	_
22	Concat Concatenation of 2 inputs along dimension 2	Concatenation	60×120×32	
23	Gapool Global average pooling	Global average pooling	1×1×32	_
24	FC 4 fully connected layer	Fully Connected	1×1×4	Weights 4×32 Bias 4×1
25	Softmax Softmax	Softmax	1×1×4	_
26	Classoutput Crossentropyex	Classification output	_	_

3.3.2 Hyper-parameter selection

Activation function: An activation function in machine learning is a mathematical operation that works with the non-linearity of the model. It works in the output of each neuron. It grips the weighted sum from the first layer produces the output and finally proceeds to the next layer. There are different forms of activation functions like Rectifier Linear Unit (ReLU), Leaky ReLU (LReLU), and Clipped ReLU(CReLU). The activation function ReLU is commonly incorporated after the convolutional layer to reduce gradient diffusion and generate better performance.

Pooling operation: Because there are many feature maps in a deep neural network, overfitting and computational issues arise. To address this problem pooling layer is performed. The two most common pooling operations in the CNN are Max pooling and Average pooling. Max pooling considers the maximum value from the subset. Some important information is ignored using Max pooling since only the strongest feature elements are considered. To eliminate this problem Average pooling is used. Average pooling takes the average value from the subset. Although Average pooling selects all the elements, they are always treated equally and provide contributions of all essential elements. It extracts features more smoothly than Max pooling.

Optimization algorithms: Algorithms known as optimizers are used to modify weight and learning rate strategies in order to minimize losses. In this paper, three different types of optimizers—"sgdm," "adam," and "rmsprop"—are used. Rmsprop provides a slow learning rate and Adam poorly tuned hyperparameters. To overcome this problem sgdm optimizer is used which has a high variance of the parameters and reduces losses [50].

Normalization selection: The input data's characteristics are scaled in the normalization layer within the range of mean o and standard 1. Cross-channel normalization adjusts local reactions for each activation by using various channels. It introduces computational complexity when the relationship between channels is complex. Batch normalization is used to enhance the problem by normalizing the features in intermediate layers of deep neural networks. It increases accuracy, increases stability, and decreases overfitting issues while training quickly. The batch normalization formula is:

$$\hat{x} = \frac{x - E[x]}{\sqrt{V_{ar}}\,[x]} \tag{3.2}$$

With the mean value set at 0 and the standard deviation at 1, batch normalization normalizes a subsequent input by subtracting the mini-batch mean and dividing it by the mini-standard deviation.

Output layer selection: Another name for a fully connected layer is a dense layer that joins all of the nodes and neurons from the uppermost layer to the bottommost layer. One vector represents all of the features from the final convolution layer. As a result of the high number of parameters, computational complexity, and the overfitting issue occur. Instead of using a fully connected layer, global average pooling is employed. Determine the average value of every feature map and generate a single value for each channel without flattening the global average pooling. It offers optimal performance, minimizes the number of parameters, and guards against overfitting [51].

Table 3.2: Chart of hyper-parameter

Parameter	Size/Value
Activation function	Leaky ReLU
Pooling operation	Average Pooling
Optimization algorithms	Sgdm
Normalization selection	Batch Normalization
Output layer selection	Global Average Pooling

3.4 Best classifier selection

The effective features of brain tumors are obtained by using the proposed DPCNN model. These features are applied as the input to various machine learning models, including Softmax, k-nearest neighbor (KNN), support vector machine (SVM), and decision tree (DT) to select the best classifier. The softmax function is inserted at the end of the output because it is when the nodes at last converge and may be classified. The softmax classifier classifies the four types of tumors. Similarly, the KNN classifier is inserted at the end of the output of the proposed DPCNN. The inputs of the KNN classifier are the effective features that ware obtained by using the proposed DPCNN. The network's parameters of KNN are given in Table 3.3. In the parameter of KNN, the N neighbors is 5, weights are "uniform", algorithm is auto, leaf_size is 30, P is 2, and metric is "minkowski".

Table 3.3: KNN classifier's parameters

KNN Parameters	Size/Value
Leaf size	30
N neighbors	5
Metric params	"none"
N jobs	"none"
Weights	"uniform"
P	2
Metric	'minkowski'
Algorithm	"auto"

The SVM classifier is also inserted at the end of the output of the proposed DPCNN model. The inputs of the SVM classifier are the effective features that ware obtained by using the proposed DPCNN model. The network's parameters of SVM are given in Table 3.4. In the parameter of SVM, the value of C is 1.0, Kerner is "rbf", degree is 3, tol value is 0.001, max_iter is –1, and decision_function_shape is "ovr".

Table 3.4: SVM classifier's parameters

SVM Parameters	Size/ Value
C	1.0
Filter/kernel	"rbf"
Gamma	"Scale"
Degree	3
Probability	"false"
Oefo	0.0
Shrin king	True
Tol	0.001
class weight	"none"
cache size	200
Max. iter	-1
Verbose	False
Random state	"none"
Decision function shape	"ovr"
Break ties	"false"

CHAPTER 4: RESULT AND ANALYSIS

This chapter analyzes the performance of the suggested system. Section 4.1 explains the experimental data. Section 4.2 describes the experimental results and analysis, including the suggested system's performance, and evaluation measurements of the proposed system.

4.1 Experimental data collection

A dataset is a gathering of images used to train or evaluate a model. This model uses datasets from the free source "Kaggle" [49]. The MRI pictures in this collection include four categories of brain tumor images: glioma (826 images), meningioma (822 images), no tumor (395 images), and pituitary (827 images). The original images of the Kaggle dataset are 256 × 256. Figure 4.1 depicts the four types of brain tumor photos in the Kaggle dataset. As a result, 80% of the image of the data are loaded for training, with 20% loaded for testing.

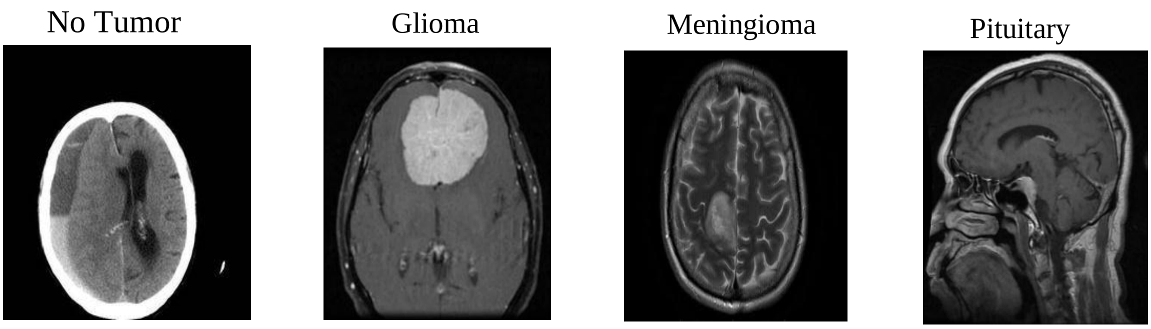


Figure 4.1: MRI images of the Kaggle dataset [49]

4.2 Experimental results and analysis

The primary objective of this proposed framework is to identify and categorize brain tumors using MRI scans of the 3459 pituitary, glioma, meningioma, and tumor-free samples in the dataset. The dataset is first resized from (256×256) to (60×60) using the image resize tool. After that, the data is added with more photos to boost the model's ability to generalize by using image augmentation. Lightweight DPCNN is used to extract features, and different classifier types—such as Softmax, KNN, and SVM—are used to classify the features. Twenty percent of the dataset is loaded for testing, and eighty percent is loaded for training. The learning rate (.001,.002,.001) and epoch (15,20,25) are used to train the model. Delivers the greatest accuracy at 97.7% at the epoch 20 learning rate of.002, as indicated by the bold letters. The experiment is run entirely inside MATLAB 2020. AMD Ryzen 5, 8GB RAM, AMD Radeon graphics, and Windows 10 Home 64-bit are included in the system specifications.

4.2.1 Experimental outcome

A table used to summarize a classification algorithm's performance is called a confusion matrix. The values of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) are provided by a confusion matrix. By employing a confusion matrix, the model's computation seeks to raise the values of TP and TN and lower the values of FP and FN. The output classes are displayed on the Y axis, while the target classes are displayed on the X axis [39]. The PCNN-Softmax, PCNN-KNN, and PCNN-SVM confusion matrices are provided by the suggested architecture accordingly.

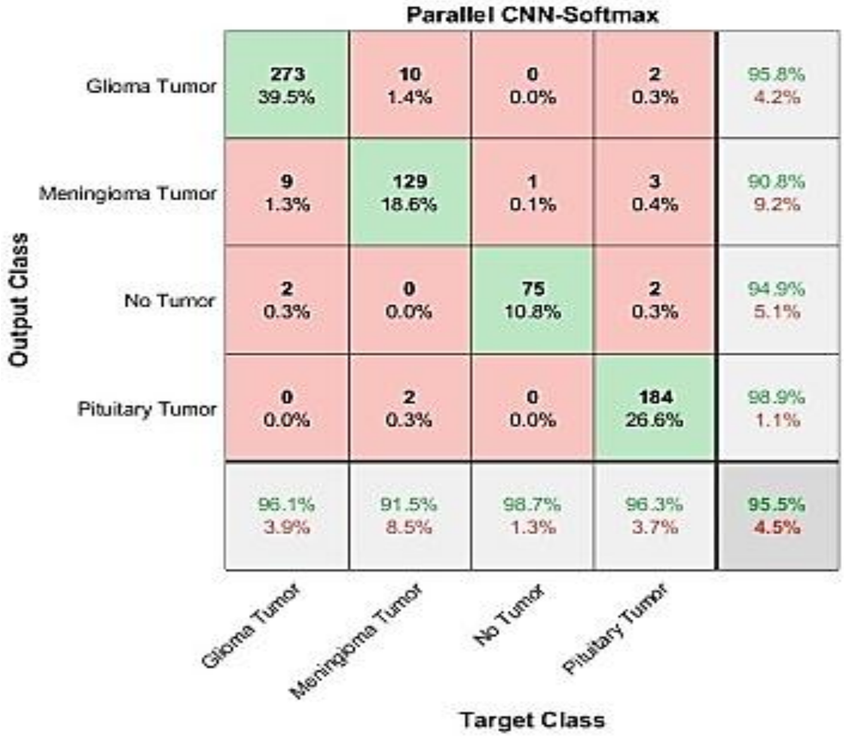


Figure 4.2: Confusion matrix of parallel CNN-Softmax

Figure 4.3: Confusion matrix of parallel CNN-KNN

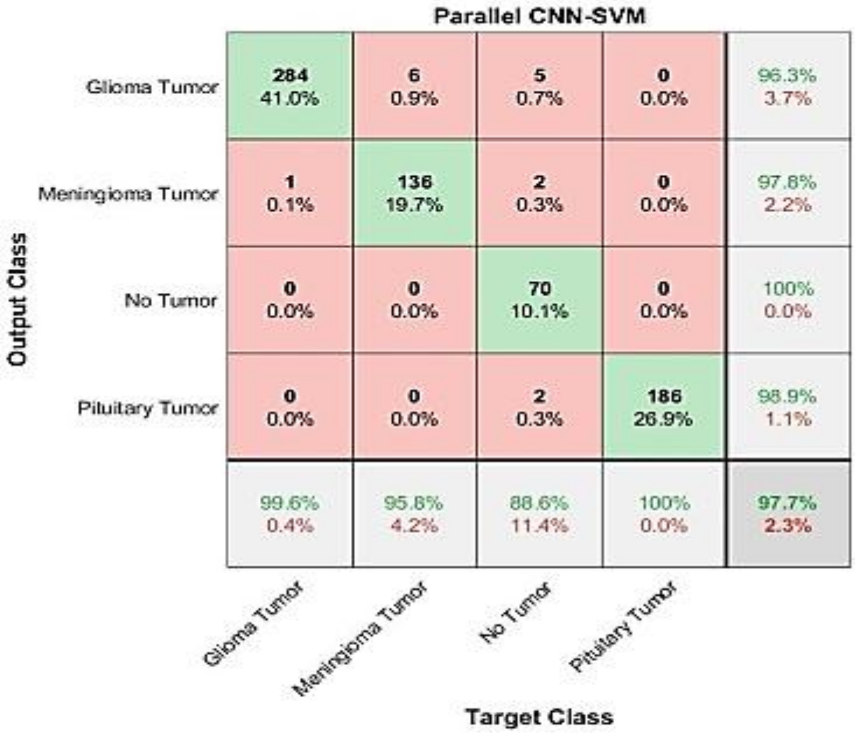


Figure 4.4: Confusion matrix of parallel CNN-SVM

Metrics used to evaluate a system, process, or outcome's efficacy are called performance metrics. The essential performance indicators, which are derived from true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN), are accuracy, sensitivity, specificity, and precision. The following are the formulae for precision, sensitivity, specificity, and accuracy [52]:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{4.1}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{4.2}$$

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{4.3}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \qquad (4.4)$$

Table 4.1: Performance analysis of classification on DPCNN-Softmax

Proposed method	TP	TN	FP	FN	Accuracy	Specificity	Sensitivity	Precision
Glioma	273	388	11	12	96.63%	97.24%	95.78%	96.12%
Meningioma	129	532	12	13	96.35%	97.79%	90.84%	91.49%
No tumor	75	586	1	4	99.24%	99.82%	94.94%	98.68%
Pituitary	184	477	7	2	98.66%	98.55%	98.92%	95.33%
Average					97.72%	98.35%	95.12%	95.40%

The confusion matrix of the DPCNN-Softmax architecture is utilized to determine the Accuracy, Specificity, Sensitivity, and Precision of pituitary tumors, meningiomas, gliomas, and no tumors utilizing TP, TN, FP, and FN, as shown in Table 4.1 For accuracy, 97.72% is the mean value, 98.35% is the mean value, for specificity, 95.12% is the mean value, for sensitivity, and 95.40% is the mean value for precision.

Table 4.2: Performance analysis of classification on DPCNN-KNN

Proposed method	TP	TN	FP	FN	Accuracy	Specificity	Sensitivity	Precision
Glioma	271	375	14	21	94.86%	96.40%	92.80%	95.08%
Meningioma	118	528	24	14	94.44%	95.65%	89.39%	83.09%
No tumor	74	572	5	2	98.92%	99.13%	97.37%	93.67%
Pituitary	183	463	3	9	98.18%	99.35%	95.31%	98.38%
Average					96.6%	97.63%	93.71%	92.56%

The DPCNN-KNN model's confusion matrix, shown in Table 4.2, is used to determine the Accuracy, Specificity, Sensitivity, and Precision of pituitary tumors, meningiomas, gliomas, and no tumors using TP, TN, FP, and FN. The average value for Accuracy is 96.6%, Specificity is 97.63%, Sensitivity is 93.71%, and Precision is 92.56%.

Table 4.3: Performance analysis of classification on DPCNN-SVM

Proposed method	TP	TN	FP	FN	Accuracy	Specificity	Sensitivity	Precision
Glioma	284	392	1	11	98.25%	99.74%	96.27%	99.64%
Meningioma	136	540	6	3	98.68%	98.90%	97.84%	95.77%
No tumor	70	606	11	0	98.40%	98.21%	100%	86.41%
Pituitary	186	490	0	2	99.70	100%	98.93%	100%
Average					98.76%	99.21%	98.26%	95.45%

To determine the Accuracy, Specificity, Sensitivity, and Precision of glioma, meningioma, no tumor, and pituitary tumor utilizing TP, TN, FP, and FN, the confusion matrix of the DPCNN-SVM design is executed above Table 4.3. The average values for Accuracy, Specificity, Sensitivity, and Precision are 98.76%, 99.21%, 98.26%, and 95.45%, respectively. In terms of performance, SVM with the DPCNN model fared better than the other two classifiers.

4.2.2 The proposed system's performance

In order to improve the accuracy of the proposed method, various kinds of parameters are selected via tuning, such as the activation function, pooling layer, normalizing layer, optimizer algorithm, and output layer. Batch normalization is selected in the normalization layer, the SGDM optimizer performs as the best optimization, the activation function yields the highest accuracy in the Leaky ReLU Layer, average pooling yields the highest accuracy in the pooling layer, and the global average layer provides the output. In conclusion, three classifiers are employed: Softmax, KNN, and SVM. Of these, SVM has the highest level of accuracy in distinguishing between the various tumor kinds.

4.2.2.1 Performance of the classifier

The features derived from the final layer of the suggested model are fed into a number of machine learning models, such as Softmax, k-nearest neighbor (KNN), and support vector machines (SVM), in order to optimize classification performance. Additionally, each model's categorization capabilities are examined separately. Figure 4.5 displays the classifier's performance. The performance of the softmax classifier is shown in blue, whereas the performance of the KNN classifier is represented in red. The performance of the SVM classifier is shown in green. The value of 97.72% is obtained for accuracy, 98.35% for specificity, 95.12% for sensitivity, and 95.40% for precision for the Softmax classifier with DPCNN. The value of 96.60% is achieved for accuracy, 97.63% for specificity, 93.71% for sensitivity, and 92.56% for precision for the KNN classifier with DPCNN. The value of 98.76% is obtained for accuracy, 99.21 for specificity, 98.26% for sensitivity, and 95.45% for precision for the SVM classifier with DPCNN. It can be observed in Figure 4.5 that the performance of the SVM classifier is much better than the performance of the softmax classifier and KNN classifier.

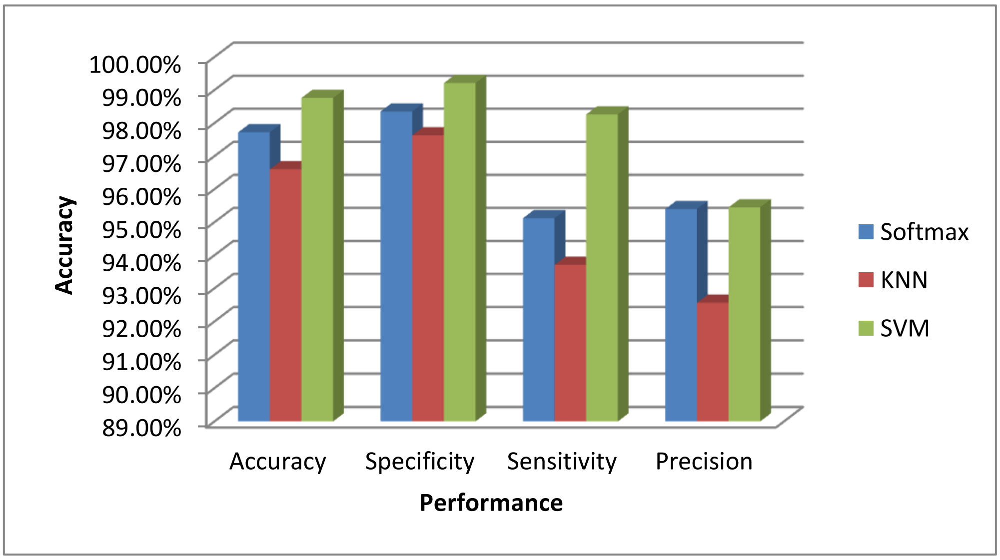


Figure 4.5: Performances of classifiers

4.2.2.2 Impact on data augmentation

To enhance the number of images, augmentation is used in Figure 4.6. Provide accuracy of 93.9% for DPCNN-Softmax, 93.2% for DPCNN-KNN, and 94.7% for DPCNN-SVM in the absence of augmented data. In contrast, the accuracy of the augmented data is 95.1% for DPCNN-Softmax, 93.4% for DPCNN-KNN, and 96.8% for DPCNN-SVM. The accuracy provided by augmented data is superior to that of non-augmented data.

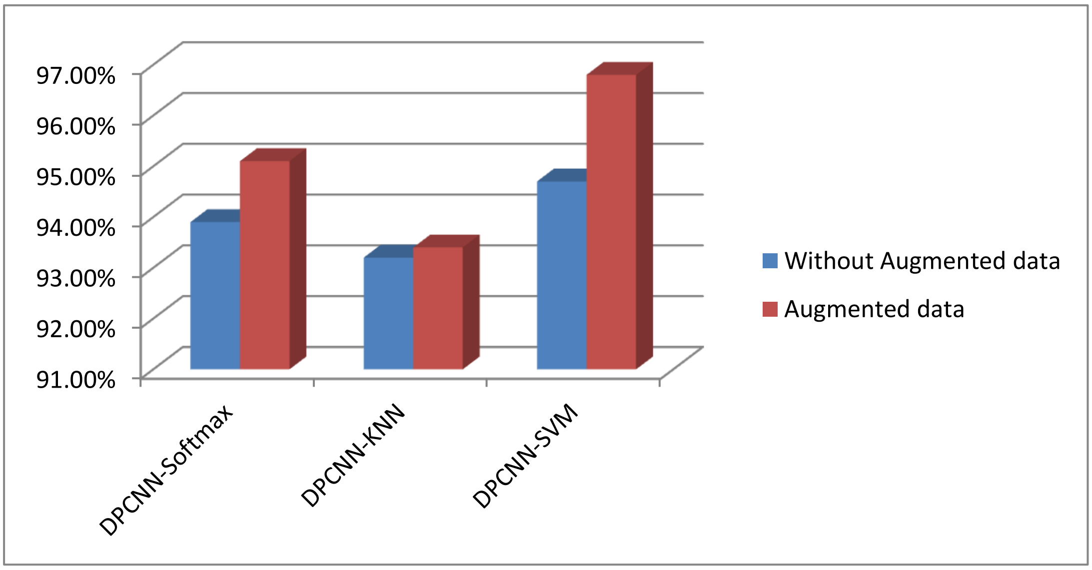


Figure 4.6: Performance analysis of classification on augmented data

4.2.2.3 Impact on activation function

The performance analysis on various activation function shapes is displayed in Table 4.4. The accuracy of the ReLU Layer activation function is 95.8% for DPCNN-SVM, 92.5% for DPCNN-KNN, and 94.4% for DPCNN-Softmax. For PCNN-Softmax, DPCNN-KNN, and DPCNN-SVM, the accuracy of the clipped Layer activation function is 93.6%, 91.1%, and 95.3%, respectively. An accuracy of 95.6% for DPCNN-Softmax, 93.2% for DPCNN-KNN, and 96.6% for DPCNN-SVM is provided by the leaky ReLU Layer activation function. When compared to other activation functions, the Leaky

ReLU Layer yields a noticeable outcome.

Table 4.4: Performance analysis of the classification of Activation function

Activation function	DPCNN-Softmax	DPCNN-KNN	DPCNN-SVM
ReLU Layer	94.4%	92.5%	95.8%
Clipped Layer	93.6%	91.1%	95.3%
Leaky ReLU Layer	**95.6%**	**93.2%**	**96.6%**

4.2.2.4 Impact on pooling layer

The performance study of the different types of pooling layers that are used to minimize the feature map is shown in Table 4.5. An accuracy of 95.3% for DPCNN-Softmax, 93.1% for DPCNN-KNN, and 96.4% for DPCNN-SVM is provided by the Max pooling layer. Conversely, with DPCNN-Softmax, DPCNN-KNN, and DPCNN-SVM, the average pooling layer offers an accuracy of 95.4%, 93.2%, and 97.0%, respectively. Among the several pooling layers, the average pooling layer guarantees the best performance.

Table 4.5: Performance analysis of classification on the Pooling layer

Pooling layer	DPCNN-Softmax	DPCNN-KNN	DPCNN-SVM
Max pooling layer	95.3%	93.1%	96.4%
Average pooling layer	**95.4%**	**93.2%**	**97.0%**

4.2.2.5 Impact on normalization layer

Performance analysis on the normalization layer, which is used to enhance performance and expedite training, is shown in Figure 4.7. The 95.1% accuracy for DPCNN-Softmax, 92.6% accuracy for DPCNN-KNN, and 96.4% accuracy for DPCNN-SVM are attained by the cross-normalization layer. In contrast, the accuracy of the batch normalization layer is 95.3% for DPCNN-Softmax, 92.8% for DPCNN-KNN, and 97.1% for DPCNN-SVM. Batch normalization produces the best outcome out of all of them.

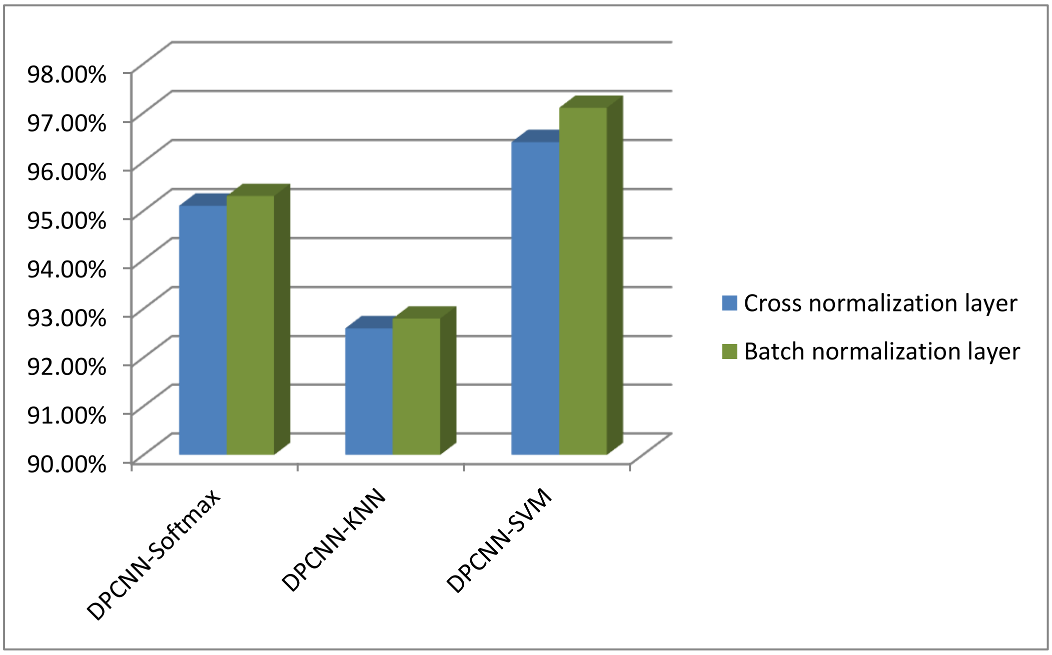


Figure 4.7: Performance analysis of classification on normalization

4.2.2.6 Impact on optimization algorithm

The optimization algorithm, which is used to reduce or maximize a function by iteratively changing the input parameters, is shown in Figure 4.8 performance analysis. The accuracy of the Adam optimizer is 95.4% for DPCNN-SVM, 93.1% for DPCNN-KNN, and 94.4% for DPCNN-Softmax. On the other hand, Rmsprop offers 92.2% accuracy for DPCNN-Softmax, 92.9% accuracy for DPCNN-KNN, and 94.9% accuracy for DPCNN-SVM. Lastly, for DPCNN-Softmax, 93.1% for DPCNN-KNN, and 97.3% for DPCNN-SVM, sgdm offers an accuracy of 95.5%. The sgdm optimizer provides the best outcome out of all of them.

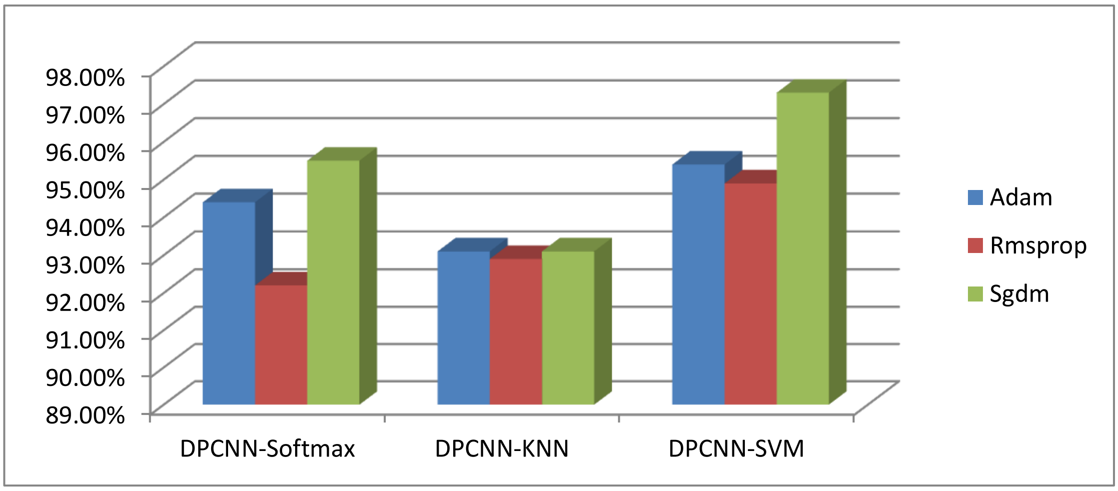


Figure 4.8: Performance analysis of classification on the Optimization

4.2.2.7 Impact on global average pooling

The performance analysis on the output layer that generates the final outputs or forecasts is displayed above in Figure 4.9. Accuracy for fully connected layers is 95.2% for DPCNN-Softmax, 93.2% for DPCNN-KNN, and 97.3% for DPCNN-SVM. Global average pooling shows an accuracy of 95.5%. for DPCNN-Softmax, 93.4% for DPCNN-KNN, and 97.7% for DPCNN-SVM, rather than having a fully connected layer, global average pooling yields a noticeable result.

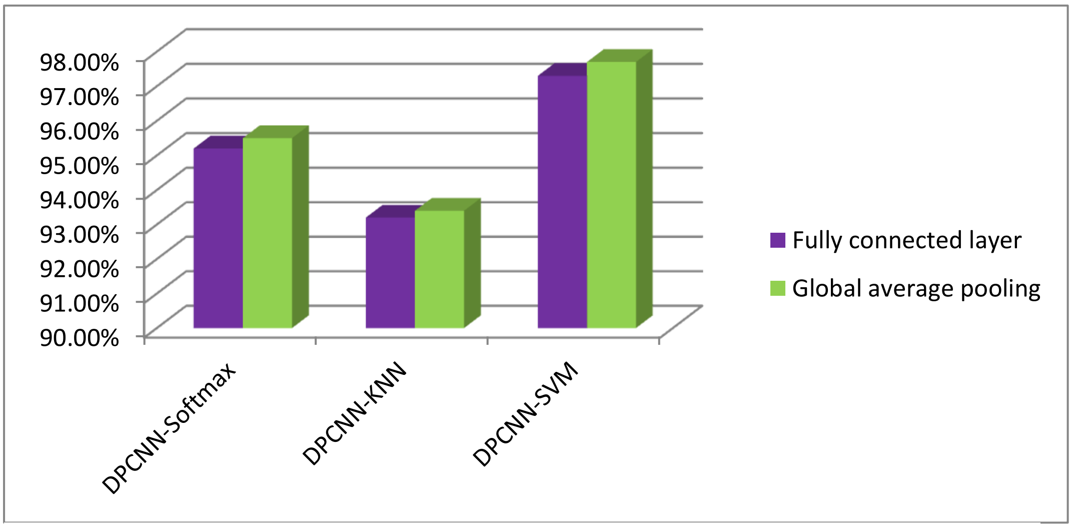


Figure 4.9: Impact of global average pooling on the output layer

4.2.3 Comparison Result

4.2.3.1 Comparison result with tuning image size

The proposed system's performance was evaluated using different image dimensions. The result of the proposed system with the different image dimensions is shown in Table 4.6. When the dimension of the image is small, the accuracy of the proposed model is equally small. The accuracy increased gradually as the size of the image was expanded. Additionally, the proposed system in 60×60 image achieved 97.7% accuracy. However, as the matrix size gets larger, the accuracy falls below the previous result. The accuracy of the 256x256 image size has been pretty low. To record specialized features, the matrix size needs to be appropriate.

Table 4.6: Comparison result with tuning image size

Image Size	Number of Epoch	Accuracy
10×10	20	95.60%
16×16	20	96.35%
32×32	20	96.50%
60×60	**20**	**97.70%**
64×64	20	97.35%
256×256	20	95.30%

4.2.3.2 Comparison result with tuning number of epoch

Different numbers of epochs were used to compare the proposed system's performance. Table 4.7 shows a comparison of the proposed system's results with the epoch number. The relatively low number of epochs carried out can be related to the limitation of increasing accuracy. The accuracy has improved as the number of epochs has increased. In the 30^{th} epoch, we reached 97.7% accuracy. After the 20^{th} epoch, the proposed system has reached saturation accuracy. The number of epochs was raised to get significantly higher accuracy with fewer epochs but takes more time.

Table 4.7: Comparison result with tuning number of epoch

Epoch	**15**	**20**	**25**
Classifier	Accuracy		
DPCNN-Softmax	93.5%	94.2%	92.4%
DPCNN-KNN	93.2%	93.8%	92.2%
DPCNN-SVM	96.4%	**97.7%**	97.7%

4.2.3.3 Comparison result with tuning learning rate

Different learning rates were used to compare the proposed system's performance. Table 4.8 shows a comparison of the proposed system's results with the learning rate. The relatively low learning rate carried out can be related to the limitation of increasing accuracy. The accuracy has improved as the number of epochs has increased. In the 0.002 learning rate, we reached 97.7% accuracy.

Table 4.8: Comparison with different values of learning rates

Learning rate	**0.001**	**0.002**	**0.003**
Classifier	**Accuracy**		
DPCNN-Softmax	91.8%	94.2%	94.9%
DPCNN-KNN	91.5%	93.8%	94.5%
DPCNN-SVM	95.2%	**97.7%**	97.0%

4.2.3.4 Comparison with pre-trained models

Several pre-trained models such as Google Net, VGG16, VGG19, ResNet18, and ResNe50 provide performance accuracy of 93.90% 94.80%, 95.30%, 96%, and 96.8% accordingly. Those models have a large number of layers, so faces are computationally complex and require a long time due to the large number of parameters. To address these issues, the suggested lightweight DPCNN-SVM architecture provides the best accuracy of 97.7% which is shown in Figure 4.10.

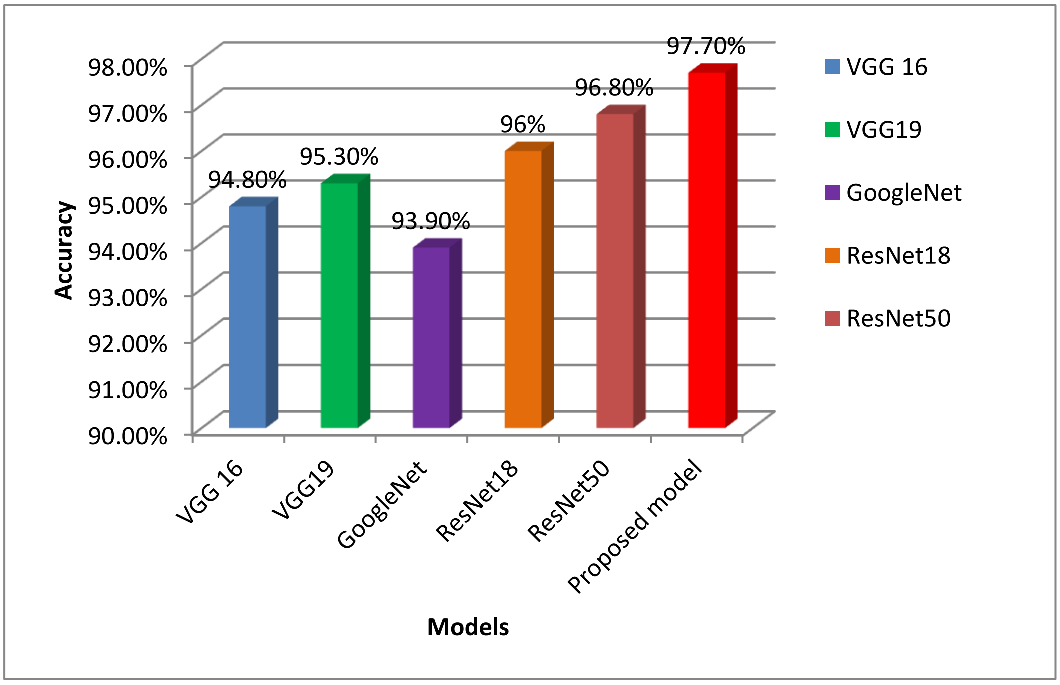


Figure 4.10: Comparison proposed model with the pre-trained model

4.2.3.5 Comparison of the proposed system Vs existing works

Deep learning networks have been used recently to identify and classify various tumor types. The suggested model, which is based on earlier studies employing different models with the same brain tumor, is displayed in Table 4.9. As compared to the earlier work, the suggested structure offers the highest accuracy. Such as Md Istyaq Mahmud [53] utilized CNN to extract features, and the model was compared to other obtained models, such as ResNet-50, VGG16, and Inception V3, and the predicted model outperformed the other models with an accuracy of 93.3%. Sajeed Ullah [54] proposed a model where features were extracted using the Gabor filter and ResNet-50 and classified by SVM individually. Next, the features from both approaches were combined and classified using SVM, with the combined features exhibiting the best accuracy of 95.7%. Dehkordi suggested a convolutional neural network was employed to classify brain images accurately. The training hyper-parameter of CNN was optimized via the Nonlinear Levy Chaotic Moth Flame Optimizer (NLCMFO). The brain images of the dataset from Harvard Medical School were used, along with the BRATS 2015 dataset. The suggested model is evaluated and contrasted using different optimization methods and the accuracy was 97.4% [55]. The proposed model, DPCNN-SVN, was utilized to extract features both locally and globally and classify them with the highest accuracy of 97.7%, in contrast to the previous research.

Table 4.9: Comparison of the proposed system Vs related works

Reference	Methods	Accuracy
Md Istyaq Mahmud [53]	CNN, ResNet50, VGG16, InceptionV3	93.3%
Sajeed Ullah [54]	Gabor filter, ResNet-50, SVM	95.7%
Dekhordi [55]	CNN, NLCMFO	97.4%
Proposed method	PDCNN-SVM	97.7%

CHAPTER 5: CONCLUSIONS AND FUTURE STUDY

This chapter analyzes the conclusion and future study. Section 5.1 explains an overall conclusion is executed by discussing the brief recap of the thesis along with the interpretations of the results. Section 5.2 describes the challenges of designing proposed. Future study recommends to analysis others models to modify are describes in section 5.3.

5.1 Conclusion

A brain tumor is a growth that affects the brain or skull and is found in the tissues around it.

Benign or malignant growth can be identified in different ways. Primary brain tumors in the brain are known as benign, whereas secondary tumors, also known as brain metastases, are usually caused by tumors outside the brain. There are many various types of brain tumors in the world, but the most well-known include glioma, meningioma, and pituitary tumors. Glioma tumors are found in glial cells with higher or lower grades, also known as astrocytes and cerebral hemispheres. Pituitary tumors originate from the pituitary gland, and meningioma tumors originate from arachnoid cap cells.

To identify and categorize different kinds of brain tumors from MRIs, ANNs are being developed. The most significant drawback of this technique is that ANN cannot extract the characteristics automatically. To address this issue a decade ago, deep learning networks were established. Certain deep machine learning models, such as CNN, have the ability to automatically extract features from MRIs. However, the disadvantage of this process is that it can arbitrarily extract unknown features, which can lead to overfitting issues.

Considering these issues, an effective DPCNN is developed that is capable of precisely detecting and classifying a wide range of tumor forms, including glioma, meningioma, no tumor, and pituitary tumors. The proposed model is designed based on the examination of the results of the experiments by tuning the hyper-parameters like epochs, learning rates, pooling layers, activation layers, optimizing algorithms, and lastly output layers. Initially, resized data is augmented by using geometric transformation operators, then augmented data is operated on by the DPCNN to extract the knowing features both locally and globally, and lastly, different kinds of classifiers like Softmax, KNN, and SVM are used to classify them. The accuracy of DPCNN-Softmax, DPCNN-KNN, and DPCNN-SVM is 95.5%, 93.4%, and 97.7%, respectively, and among them, lightweight DPCNN-SVM provides the highest accuracy of 97.7%. The proposed framework, compared with the pre-trained model, has the least amount of computing parameters and time.

In the future, would like to expand studies on the other types of cancers like stomach cancer, liver cancer, breast cancer, etc.

5.2 Limitations and challenges

The application of deep learning algorithms for brain tumor detection has been widely used due to their many benefits. However, DL algorithms have some limitations and some challenges for brain tumor detection and classification which are listed below:

- ❖ When fed large amounts of images from a dataset, deep learning algorithms perform excellently. As a result, gathering large amounts of brain tumor data for deep learning algorithms and diagnosis is a huge difficulty.

- ❖ Training deep learning models is an expensive and time-consuming procedure. The most complex models require weeks of training on high-performance GPU workstations. Additionally, a large amount of memory is necessary to train the model.
- ❖ Once more, unbalanced data sets make deep learning less effective. It incorrectly classifies the class with more data samples. Having an MRI brain tumor dataset is thus a big issue in and of itself. Therefore, alternative methods should be employed to train deep learning models when the images of the dataset are unbalanced.
- ❖ It is challenging for brain tumor detection to arise from the variations in tumor location, shape, and size.

5.3 Future recommendations

- ❖ In the future, I would like to expand studies on the other types of cancers like stomach cancer, liver cancer, breast cancer, etc.
- ❖ The suggested work will be developed further in the future to employ a 3D structure for 3D brain MRI image-based tumor identification.
- ❖ Five pre-trained models are utilized in this research. It can be extended to several pre-trained models.
- ❖ Three classifiers are used in this research. It can be extended to several classifiers.

References

1. Sadoon, Toga A., and Mohammed H. Ali. "Deep learning model for glioma, meningioma and pituitary classification." Int. J. Adv. Appl. Sci. ISSN 2252, no. 8814 (2021): 8814.P.K. Kankar, Satish C. Sharma, S.P. Harsha, “Rolling element bearing fault diagnosis using wavelet transform”, *Elsevier*, *Neurocomputing* 74 (2011) 1638–1645.
2. Grisold, Wolfgang, and Anna Grisold. "Cancer around the brain." Neuro-oncology practice 1, no. 1 (2014): 13-21.
3. https://www.virtualtrials.org/symptoms.cfm?gclid, Access time, 10th May 2024
4. www.hopkinsmedicine.org/health/conditions-and-diseases/ gliomas (accessed on 12 February 2023).
5. Kamireddy Rasool Reddy, Ravindra Dhuli, A novel lightweight CNN Architecture for The Diagnosis of Brain tumors using MR images.2023,10.3390.
6. M. H. Kabir, M. S. Islam, M. Mohiuddin, and M. Golam Mortuza, "Fuzzy-Based Smart Path Guidance System for Visually Impaired and Deaf Person," 2021 IEEE 4th International Conference on Computing, Power and Communication Technologies (GUCON), Kuala Lumpur, Malaysia, 2021, pp. 1-5, doi: 10.1109/GUCON50781.2021.9573762.
7. Rogers, L., Barani, I., Chamberlain, M., Kaley, T.J., McDermott, M., Raizer, J., Schiff, D., Weber, D.C. Wen, P.Y. Vogelbaum, M.A. Meningiomas: Knowledge base, treatment outcomes, and uncertainties. A RANO review. J. Neurosurg. 2015, 122, 4–23.
8. P. Thejaswini, MB. Bhat, and MK. Prakash, “Detection and classification of tumor in brain MRI, International Journal of Engineering and Manufacturing (IJEM), vol.9, no1, pp 11-20,2019.
9. El-Dahshan, E.-S.A. Hosny, T. Salem, A.-B.M. Hybrid intelligent techniques for MRI brain images classification. Digit. Signal Process. 2010, 20, 433–441.
10. G Hemanth, M Janardhan, L Sujihelen; Design and implementing brain tumor detection using machine learning approach,2019,1289-1294.

11. Sajjad M, Khan S, Muhammad K, Wu W, Ullah A, Baik SW. (2019). Multi-grade brain tumor classification using deep CNN with extensive data augmentation. Journal of computational science, 30(1), 174-82.
12. Sultan, H.H. Salem, N.M.Al-Atabany, W. Multi-Classification of Brain Tumor Images Using Deep Neural Network.2019, 7, 69215–69225.
13. Garg, G. Garg, R. Brain tumor detection and classification based on hybrid ensemble classifier. arXiv 2021, arXiv:2101.00216.
14. Raza, A. Ayub, H. Khan, J.A. Ahmad, I.SSalama, A. Daradkeh, Y.I.Javeed, D. Ur Rehman, A. Hamam, H. A Hybrid Deep Learning-Based Approach for Brain Tumor Classification. Electronics 2022, 11, 1146.
15. Afshar, P. Plataniotis, K.N.Mohammadi, A Capsule Networks for Brain Tumor Classification Based on MRI Images and Coarse Tumor Boundaries; ICASSP 2019—2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 12–17 May 2019;1368–1372.
16. Mohamed Ali Habib, Brain Tumor Detection Using Convolution Neural Network,25 November,2021.
17. https://www.aans.org/en/Patients/Neurosurgical-Conditions-andTreatments/ Brain Tumors, Access time, 10th May 2024.
18. Hagargi, A.P. Shubhangi, D.C. Brain tumor detection and ART classification technique in MR brain images using RPCA QT decomposition; 2018, 5.
19. Yamashita, Rikiya, Mizuho Nishio, Richard Kinh Gian Do, and Kaori Togashi. "Convolutional neural networks: an overview and application in radiology." Insights into imaging 9 (2018): 611-629.
20. Sejuti, Zarin Anjuman, and Md Saiful Islam. "An efficient method to classify brain tumor using CNN and SVM." In 2021 2nd International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST), pp. 644-648. IEEE, 2021.
21. Jyotsana (2023) Image augmentation techniques, Medium. Available at: https://medium.com/@jyotsana.cg/image-augmentation-techniques798243f6afdf (Accessed: 06 September 2024).
22. C. Saravanan, " Color Image to Grayscale Image Conversion, " 2010

Second International. Conference on Computer Engineering and Applications, Bali, Indonesia, 2010, pp. 196-199, doi:10.1109/ICCEA.2010.192.

23. https://www.mathworks.com/support/search.html?q=image%20resize%20procedure&page=1.
24. M. Mohiuddin, M. S. Islam and M. H. Kabir, "Performance Analysis of Bearing fault diagnosis using Convolutional Neural Network," 2021 IEEE 4th International Conference on Computing, Power and Communication Technologies (GUCON), Kuala Lumpur, Malaysia, 2021, pp. 1-6, doi: 10.1109/GUCON50781.2021.9573710.
25. Mohiuddin, M.; Islam, M.S. Rolling Element Bearing Faults Detection and Classification Technique Using Vibration Signals. Eng. Proc. 2022, 27, 53. https://doi.org/10.3390/ecsa-9-13339.
26. Mohiuddin, M.; Islam, M.S.; Islam, S.; Miah, M.S.; Niu, M.-B. Intelligent Fault Diagnosis of Rolling Element Bearings Based on Modified AlexNet. Sensors 2023, 23, 7764. https://doi.org/10.3390/s23187764.
27. Zhang, W.; Li, C.; Peng, G.; Chen, Y.; Zhang, Z. "A deep convolutional neural network with new training methods for bearing fault diagnosis under noisy environment and different working load". Mech. Syst. Signal Process. 2018, 100, 439–453.
28. Goodfellow, I.; Bengio, Y.; Courville, A. Deep Learning; MIT Press: Cambridge, MA, USA, 2016. [Accessed: 17-January-2023].
29. Zhang, W.; Peng, G.; Li, C. "Bearings fault diagnosis based on convolutional neural networks with 2-D representation of vibration signals as input". In Proceedings of the 2016 the 3rd International Conference on Mechatronics and Mechanical Engineering (ICMME 2016), Shanghai, China, 21 23 October 2016; pp.1–5.
30. Lecun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. "Gradient-based learning applied to document recognition". Proc. IEEE 1998, 86, 2278–2324.
31. https://www.sciencedirect.com/topics/engineering/convolutional,layer#:~:text=A2convolutional2lay20map. [Access: 16 july 2024]
32. Nair, V. Hinton, G.E. Rectified linear units improve Restricted Boltzmann

machines. In Proceedings of the 27th International Conference on Machine Learning (ICML-10), Haifa, Israel, 21–24 June 2010.

33. Montavon, G.; Samek, W.; Müller, K.-R. Methods for interpreting and understanding deep neural networks. Digit. Signal Process. 2018, 73, 1–15.
34. https://www.jait.us/index.php?m=content&c=index&a=show&catid=208&id=1174#:~:text=The%20parallel%20CNNs%20can%20have,feature%20extraction%20can%20be%20improved. [Access: 16 july 2024]
35. https://www.linkedin.com/advice/0/what-some-most-popular-widely-used-pre-trained. [Accessed: 15-January-2023]
36. c. szegedy, w. liu, y. jia, et al. going deeper with convolutions. arxiv preprint arxiv: 1409.4842,2014.
37. Guo, Z.; Chen, Q.; Wu, G.; Xu, Y.; Shibasaki, R.; Shao, X. "Village Building Identification Based on Ensemble Convolutional Neural Networks", Sensors 2017, 17, 2487. https://doi.org/10.3390/s17112487.
38. Xiangyu Zhang, Jianhua Zou, Kaiming He, Jian Sun, "Accelerating Very Deep Convolutional Networks for Classification and Detection", https://arxiv.org/abs/1505.06798.
39. https://www.sciencedirect.com/topics/computer-science/vgg-19-convolutional-neural-network [Access: 16 july 2024].
40. https://www.wonderbreadcaribbean.net/?y=vgg-19-convolutional-neural network- all-about- machine-learning EVr45AUm[Access: 16 july 2024]
41. www.sciencedirect.com/topics/computer-science/residual-neural network ResNet18 2Dlayer,convolutional%20layers%20to%20function%20efficiently [Access: 16 july 2024].
42. Misra, S.; Jeon, S.; Lee, S.; Managuli, R.; Jang, I.-S.; Kim, C. Multi-Channel Transfer Learning of Chest X-ray Images for Screening of COVID-19. Electronics 2020, 9, 1388. https://doi.org/10.3390/electronics9091388.
43. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.
44. E. Rezende, G. Ruppert, T. Carvalho, F. Ramos and P. de Geus, "Malicious

Software Classification Using Transfer Learning of ResNet-50 Deep Neural Network," 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA), Cancun, Mexico, 2017, pp. 1011-1014, doi: 10.1109/ICMLA.2017.00-19.

45. https://c3.ai/glossary/data-science/classifier [Accessed: 15-July-2023].
46. Muhammad Asad Iqbal Khan, "Introduction to Softmax Classifier in PyTorch", Deep Learning with PyTorch, January 1, 2023 [Accessed: 05-August-2023].
47. Vijay Wasule, Poonam Sonar; Classifiacation of Brain MRI Using SVM and KNN Classifier;2017, IEEE 3rd International conference on Sensing, Signal Processing and Security (ICSSS).
48. Álvarez-Alvarado, J.M.; Ríos-Moreno, J.G.; Obregón-Biosca, S.A.; Ronquillo-Lomelí, G.; Ventura-Ramos, E., Jr.; Trejo-Perea, M. Hybrid Techniques to Predict Solar Radiation Using Support Vector Machine and Search Optimization Algorithms: A Review. Appl. Sci. 2021, 11, 1044. https://doi.org/10.3390/app11031044.
49. https://www.kaggle.com/sartajbhuvaji/brain-tumor-classification mri?select=Training. [Access: 16 july 2024].
50. https://medium.com/mlearning-ai/optimizers-in-deep-learning-7bf81fed78a0
51. Nair, V. Hinton, G.E. Rectified linear units improve Restricted Boltzmann machines. In Proceedings of the 27th International Conference on Machine Learning (ICML-10), Haifa, Israel, 21–24 June 2010.
52. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition.3rd International Conference on Learning Representations, ICLR, San Diego, CA, USA, 7–9 May 2015; pp. 1–14.
53. Md Istyaq Mahmud, Muntasir Mamun, Ahmed Abdelgawad; A Deep Analysis of Brain Tumor Detection from MR Images Using Deep Learning Networks.23 March 2023; https;//doi.org/10.3390/a16040176.
54. Sajeed Ullah, Mehran Ahmad, Shahzad Anwar, Muhammad Irfan Khattak; An Intelligent Hybrid Approach for brain tumor detection; http://doi.org/10.51846/vol6iss1pp34-42.
55. Dehkordi, A.A.; Hashemi, M.; Neshat, M.; Mirjalili, S.; Sadiq, A.S. Brain Tumor

Detection and Classification Using a New Evolutionary Convolutional Neural Network. arXiv 2022, arXiv:2204.12297.

56. Kheirollahi, Majid, Sepideh Dashti, Zahra Khalaj, Fatemeh Nazemroaia, and Parvin Mahzouni. "Brain tumors: Special characters for research and banking." Advanced biomedical research 4, no. 1 (2015): 4.
57. Lukoff, Jonathan, and Jaime Olmos. "Minimizing medical radiation exposure by incorporating a new radiation “vital sign” into the electronic medical record: quality of care and patient safety." The Permanente Journal 21 (2017).
58. Vankdothu, Ramdas, and Mohd Abdul Hameed. "Brain tumor MRI images identification and classification based on the recurrent convolutional neural network." Measurement: Sensors 24 (2022): 100412.
59. Yamashita, Rikiya, Mizuho Nishio, Richard Kinh Gian Do, and Kaori Togashi. "Convolutional neural networks: an overview and application in radiology." Insights into imaging 9 (2018): 611-629.
60. Sejuti, Zarin Anjuman, and Md Saiful Islam. "An efficient method to classify brain tumor using CNN and SVM." In 2021 2nd International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST), pp. 644-648. IEEE, 2021.
61. Afshar, Parnian, Konstantinos N. Plataniotis, and Arash Mohammadi. "Capsule networks for brain tumor classification based on MRI images and coarse tumor boundaries." In ICASSP 2019-2019 IEEE international conference on acoustics, speech and signal processing (ICASSP), pp.
62. Sultan, Hossam H., Nancy M. Salem, and Walid Al-Atabany. "Multi-classification of brain tumor images using deep neural network." IEEE access 7 (2019): 69215-69225.
63. Rahman, Takowa, and Md Saiful Islam. "Mri brain tumor classification using deep convolutional neural network." In 2022 International Conference on Innovations in Science, Engineering and Technology (ICISET), pp. 451-456. IEEE, 2022.