MRI Brain Tumor Detection and Classification Using Parallel Deep Convolutional Neural Network

By

Takowa Rahman, MSc Eng

20METE007P

A thesis submitted in partial fulfillment of the requirements for the degree of MASTER of SCIENCE in ELECTRONICS AND TELECOMMUNICATION ENGINEERING

Department of Electronics and Telecommunication Engineering

CHITTAGONG UNIVERSITY OF ENGINEERING AND TECHNOLOGY

March 2024

APPROVAL

The thesis titled "MRI Brain Tumor Detection and Classification Using Parallel Deep Convolutional Neural Network" submitted by Takowa Rahman, ID 20METE007P, Session: 2020-21 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 25th March 2024.

Board of Examiners

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 Head, Dept. of ETE, CUET	Member
4.	Prof. Dr. Md. Jahedul Islam Professor, Dept. of ETE, CUET	Member
5.	Prof. Dr. Muhammad Ahsan Ullah 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.

Takowa Rahman

20METE007P

Department of Electronics and Telecommunication Engineering

Chittagong University of Engineering & Technology (CUET)

Dedication

Dedicated to

My beloved parents, and **Dr. Md. Saiful Islam** sir

For their continuous support

List of Publications

1. **T. Rahman, M. S. Islam**, "MRI-Based Brain Tumor Classification Using Dilated Parallel Deep Convolutional Neural Network with Ensemble of Machine Learning Classifiers", *IEEE Access*, 2024 [**Q1 journal, Submitted**].
2. **T. Rahman, M. S. Islam**, "MRI Brain Tumor Detection and Classification Using Parallel Deep Convolutional Neural Networks", *Measurement: Sensors*, Volume: 26, Issue, Pages 100694, April 2023, doi.org/10.1016/j.measen.2023.100694 **[Q3 journal, Scopus index, Citescore: 2.7]**
3. **T. Rahman, M. S. Islam**, "MRI Brain Tumor Classification Using Deep Convolutional Neural Network", *2022 3rd International Conference on Innovations in Science, Engineering and Technology (ICISET)*, Chittagong, 26-27 February 2022, Pages 451-456, doi: 10.1109/ICISET54810.2022.9775817. **[Scopus index]**
4. **T. Rahman, M. S. Islam, "**Brain Tumors Classifications from MRI Images Using Morphological Reconstruction and Convolutional Neural Network", *International Conference on Big Data, IOT and Machine Learning*, 23-25 September, 2021.

Approval by the Supervisor

This is to certify that **Takowa Rahman** has carried out this research work under my supervision, and that she has fulfilled the relevant Academic Ordinance of the Chittagong University of Engineering and Technology, so that she is qualified to submit the following Thesis in the application for the degree of MASTER of SCIENCE in Electronics and Telecommunication Engineering (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 and foremost, I'd like to thank my GOD for making things easier than I could ever have imagined. I would like to keep in mind a few very essential people without whom I would be unable to accomplish this thesis. I would like to express my gratitude to my distinguished supervisor, Dr. Md. Saiful Islam, Associate Professor, Department of ETE, CUET, for his contributions, inspiration, and motivation. Throughout my journey, he has provided regular instruction, assistance, and motivation in overcoming various technical challenges. I'd like to thank Prof. Dr. Azad Hossain, Head, ETE, CUET, and my colleagues for their ongoing encouragement and drive to accomplish my work. I'm grateful to my family for inspiring me to pursue higher education in the first place. Their unwavering support and encouragement were the first things that prepared the path for this journey. Last but not least, I would like to acknowledge everyone at the Department of Electronics and Telecommunication Engineering (ETE), CUET, for their technical and administrative assistance with my thesis work.

.

Abstract

Brain tumors are frequently classified with high accuracy using convolutional neural networks (CNNs) and better comprehend the spatial connections among pixels in complex pictures. Due to their tiny receptive fields, the majority of deep convolutional neural network (DCNN)-based techniques overfit and are unable to extract global context information from more significant regions. While dilated convolution retains data resolution at the output layer and increases the receptive field without adding computation, stacking several dilated convolutions has the drawback of producing a grid effect. To handle gridding artifacts and extract both coarse and fine features from the images, this research suggests using a dilated parallel deep convolutional neural network (PDCNN) architecture that preserves a wide receptive field. To reduce complexity, initially, input images are resized and then grayscale transformed. Data augmentation has since been used to expand the number of datasets. Dilated PDCNN makes use of the lower computational overhead and contributes to the reduction of gridding artifacts. By contrasting various dilation rates, the global path uses a low dilation rate (2, 1, 1), while the local path uses a high dilation rate (4, 2, 1) for decrement even numbers to tackle gridding artifacts and extract both coarse and fine features from the two parallel paths. Using three different types of MRI datasets, the suggested dilated PDCNN with the average ensemble method performs better. The accuracy provided by the Multiclass Kaggle dataset-III, Figshare dataset-II, and Binary tumor identification dataset-I is 98.35%, 98.13%, and 98.67%, respectively. In comparison to state-of-the-art techniques, the suggested structure improves results by extracting both fine and coarse features, making it efficient.

বিমূর্ত

মস্তিষ্কের টিউমারগুলিকে প্রায়শই কনভোলিউশনাল নিউরাল নেটওয়ার্ক (সিএনএন) ব্যবহার করে উচ্চ নির্ভুলতার সাথে শ্রেণিবদ্ধ করা হয় এবং জটিল ছবিতে পিক্সেলগুলির মধ্যে কানেকশনগুলি আরও ভালভাবে বোঝা যায় । তাদের ক্ষুদ্র রিসিপটিভ ফিল্ডগুলির কারণে, বেশিরভাগ গভীর কনভোলিউশনাল নিউরাল নেটওয়ার্ক (ডিসিএনএন) ভিত্তিক কৌশলগুলি ওভারফিটিং সমস্যার সম্মুখীন হয় এবং আরও উল্লেখযোগ্য এরিয়া থেকে গ্লোবাল ইনফরমেশন বের করতে অক্ষম। প্রসারিত কনভলিউশন আউটপুট স্তরে ডেটা রেজোলিউশন বজায় রেখে স্তর সংযোজন এবং গণনা না করে রিসিপটিভ ফিল্ড বৃদ্ধি করে , বেশ কয়েকটি প্রসারিত কনভোলিউশন স্ট্যাক গ্রিড ইফেক্ট তৈরি করে। গ্রিডিং আর্টিফ্যাক্টগুলি ওভারকাম করতে এবং চিত্রগুলি থেকে মোটা এবং সূক্ষ্ম উভয় বৈশিষ্ট্য বের করতে, এই গবেষণাটি একটি প্রসারিত সমান্তরাল গভীর কনভোল্যুশনাল নিউরাল নেটওয়ার্ক (পিডিসিএনএন) আর্কিটেকচার ব্যবহার করার পরামর্শ দেয় যা একটি বিস্তৃত রিসিপটিভ ফিল্ড বজায় রাখে। জটিলতা কমাতে, প্রাথমিকভাবে, ইনপুট চিত্রগুলি পুনরায় আকার দেওয়া হয় এবং তারপরে গ্রেস্কেল এ রূপান্তরিত করা হয়। ডেটাসেটের সংখ্যা প্রসারিত করতে ডেটা বর্ধন ব্যবহৃত হয়েছে। প্রসারিত পিডিসিএনএন নিম্ন কম্পিউটেশনাল ওভারহেড ব্যবহার করে এবং গ্রিডিং আর্টিফ্যাক্টগুলি হ্রাসে অবদান রাখে। বিভিন্ন প্রসারণ হার তুলনা করে, গ্লোবাল পথ একটি নিম্ন প্রসারণ হার (২, ১, ১) ব্যবহার করে, যখন লোকাল পথ একটি উচ্চ প্রসারণ হার (৪, ২, ১) ব্যবহার করে গ্রিডিং আর্টিফ্যাক্টগুলি মোকাবেলা করতে এবং দুটি সমান্তরাল জোড় সংখ্যা পথ থেকে মোটা এবং সূক্ষ্ম উভয় বৈশিষ্ট্য বের করার জন্য। তিনটি ভিন্ন ধরণের এমআরআই ডেটাসেট ব্যবহার করে, প্রস্তাবিত প্রসারিত পিডিসিএনএন এভারেজ এনসেম্বল পদ্ধতির সাথে আরও ভাল পারফর্ম করে। মাল্টিক্লাস ক্যাগল ডেটাসেট -৩, ফিগশেয়ার ডেটাসেট -২ এবং বাইনারি টিউমার ডেটাসেট-১ দ্বারা প্রদত্ত যথার্থতা শনাক্তকরণ যথাক্রমে ৯৮.৩৫%, ৯৮.১৩% এবং ৯৮.৬৭% । অত্যাধুনিক কৌশলগুলির তুলনায়, প্রস্তাবিত কাঠামো সূক্ষ্ম এবং মোটা উভয় বৈশিষ্ট্য আহরণ করে ফলাফল উন্নত করে, কাঠামোকে দক্ষ করে তোলে।

Table of Contents

List of Figures

List of Tables

Nomenclature

CNN	Convolution Neural Network
DCNN	Deep Convolution Neural Network
PDCNN	Parallel Deep Convolution Neural Network
MRI	Magnetic Resonance Imaging
CT	Computed Tomography
CRF	Conditional Random Field
NN	Neural Network
FNN	Feed-forward Neural Network
NLP	Natural Language Processing
FC	Fully Connected
1D	One Dimensional
ReLU	Rectified Linear Unit
GD	Gradient Descent
CE	Cross-Entropy
SVM	Support Vector Machine
K-NN	K-Nearest Neighbor
NB	Naïve Bayes
DF	Dilation Factor
BN	Batch Normalization
PET	Positron Emission Tomography

Chapter 1: INTRODUCTION

This chapter explains the overview of recent state of the deep learning models in brain tumor detection and classification. In the field of brain tumor identification and categorization research, Section 1.1 provides the background of this research. The problem statement is then covered in section 1.2. In sections 1.3 and 1.4, the aims and objectives, and the contribution of the research are discussed, respectively.Finally, section 1.5 includes an outline of the remaining chapters of the Thesis.

1.1 BACKGROUND

The growth that may adversely impact a person's life is a brain tumor, which can appear in the tissues enclosing the brain or skull. Two characteristics can identify a benign or malignant growth. While secondary tumors, also referred to as brain metastasis tumors, and are typically formed from tumors outside the brain, primary cancers start inside the brain. Meningioma, pituitary adenomas, and gliomas are the three most common primary brain tumors. The brain, and spinal cord membrane layers, are the origin of meningioma, a tumor that grows slowly. Cancerous cells that arise in the pituitary gland are referred to as pituitary adenomas [1]. The brain tissue is compressed by the irregular growth of these tumors. Malignant tumors, in comparison with benign tumors, grow unevenly and damage the tissues around them.

Surgical techniques are frequently employed in the treatment of brain tumors [2]. Because MRI is non-interfering, it is preferred over computed tomography (CT), positron emission tomography (PMT), and X-rays [3]. It is estimated that 79,340 Americans aged 40 and older will be diagnosed with a primary brain tumor by 2023. It is estimated that one million Americans suffer from primary brain tumors; of these, 72% are benign tumors and 28% are malignant. The adults with primary brain tumors typically have meningioma

(46.1%), glioblastoma (16.4%), and pituitary tumors (14.5%) [4, 5]. Biopsies are taken for analysis after the tumor is found using standard medical techniques like MRI. The first test used in medicine to find cancer is the MRI [6, 7]. Two MRI pictures of two distinct brains are shown in Fig. 1.1.

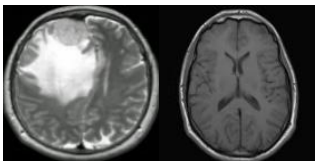


Fig. 1.1 MRI scans are performed on two different brains. On the left is a tumor, and on the right is a healthy [8].

As the number of patients has grown, individually analyzing these images has become laborious, disorganized, and frequently incorrect. A computer-aided diagnostic technique that concludes the expense of brain MRI identification needs to be developed to ease this limitation. Many attempts have been made to create an extremely effective and trustworthy method for classifying brain tumors automatically. Conventional approaches to machine learning rely on handmade qualities, which increases the cost and limits the durability of the solution. But occasionally, models of supervised learning can perform better than unsupervised learning strategies, leading to an overfitted framework that is inappropriate for another large database. These challenges underscore the significance of creating a machine learning-based, fully automated system for classifying brain tumors. CNN's architecture is based on a neural network known as the deep learning model, which excels at image recognition and classification [9, 10].

The receptive field in CNN is too tiny to produce excellent precision [11]. The size of the input region that generates the feature is referred to as the receptive field in a deep learning context. The receptive field is significant because it provides the information that the neuron has the ability to receive when generating a prediction. The size of the convolutional layer kernel and the convolution's stride define a neuron's receptive field within a CNN. A larger

receptive field is produced by a larger stride or kernel size, which gives the neuron get to more information from the input image. A smaller kernel size or a smaller stride results in a smaller receptive field, which means that the neuron has access to less information from the input image. The size of the receptive field can be adjusted to suit the specific requirements of the task at hand. The receptive field's size can be changed to meet the particular needs of the current task.

A large receptive field of the convolution kernel would help enhance the efficiency of the classification techniques because the fixed size of the sliding window in CNN misses out on utilizing techniques like convolution, pooling, and flattening. The recommended model's parameters possess the ability to acquire characteristics extracted from the images. While hyper parameters are focused on, recent iterations of CNN models have yet to focus much on them. Another important consideration is CNN's local feature collection. Furthermore, because of the limited quantity of the kernel, sharply raising the dilation rate could exacerbate feature collection failures and hinder small object detection [12]. High dilation rates may impact tiny object detection. As a result, the dilation rate has been gradually decreased in this suggested model. By doing this, the dilated feature map's sparsity has decreased and more data can be extracted from the investigated region.

Using publicly available Kaggle and Figshare datasets, this work aims to develop a fully autonomous dilated PDCNN with an average ensemble model for brain tumor classification [8, 13, 14]. This article suggests an architecture for the detection and classification of brain tumors that consists of two synchronously dilated DCNNs. Because convolutions are accurate and time-efficient processes, the dilated PDCNN with an average ensemble model performs more quickly than the conditional random field (CRF)-based methods. The recommended dilated PDCNN with an average ensemble framework incorporates batch normalization to normalize the results of previous layers.

By simultaneously integrating two DCNNs with two distinct window sizes, parallel pathways enable the model to learn both global and local features. While maintaining a large receptive field, this research also recommends managing gridding artifacts and extracting both coarse and fine characteristics from the image.

The motivations for this study are to find an effective dilated parallel deep convolutional neural network with average ensemble method to detect and classify brain tumor along with pre-processing technique that overcomes the limitations of the deep CNN model.

1.2 PROBLEM STATEMENT

Prior to beginning treatment, the most significant challenge is identifying and categorizing brain MRI tumors. There are not many studies on tumor diagnosis as a time-saving method, despite the majority of brain tumor identification research focusing on tumor slicing and positioning methods. Most DCNN-based methods are unable to acquire global context details of larger regions because of the small receptive fields. Staking multiple dilated convolutions has the disadvantage of creating a grid effect, even though dilated convolution maintains data resolution at the output layer and expands the receptive field without incorporating calculation. If the dilation factor (DF) is low, the model may have a smaller receptive field but misses the coarse characteristics. In contrast, when the DF is excessive, the model is unable to learn from the finer details.

1.3 AIMS AND OBJECTIVES

The main objectives of this research is to develop a dilated PDCNN architecture that maintains a large receptive field to cope with gridding distortions and capture both coarse and file attributes from images.

The objectives of this research are specified as below:

i. To develop an effective data pre-processing technique for brain tumor classification.
ii. To develop a dilated PDCNN with even-numbered dilation rate decrements at the local and global paths and combine two parallel CNNs with hyper-parameter tuning for brain tumor classification.
iii. To employ a feature fusion technology that significantly enhances the dynamic properties offered by the two simultaneous convolutional layers.
iv. To employ an average ensemble technique to classify the brain tumor images using obtained features.
v. To perform a comparative analysis of the proposed system to study the improvements in efficiency and performance.

1.4 CONTRIBUTION

Since brain tumor images contain multi-scale features like local, global, coarse, and fine features however, traditional CNN collects features randomly without knowing the feature types. The majority of deep CNN-based techniques cause overfitting problem. Due to their tiny receptive fields, CNN are unable to extract global context information from more significant regions. Using high dilation rates results in the gridding phenomenon that prevents the model from learning from finer features resulting in the low accuracy. To handle gridding artifacts and extract both coarse and fine features from the images, this research suggests using a dilated parallel deep convolutional neural network (PDCNN) architecture that preserves a wide receptive field. To reduce complexity, initially, input images are resized and then grayscale transformed. Data augmentation has since been used to expand the number of datasets. Dilated PDCNN makes use of the lower computational overhead and contributes to the reduction of gridding artifacts. By contrasting various dilation rates, the global path uses a low dilation rate (2, 1, 1), while the local path uses a high dilation rate (4, 2, 1) for decremented

even numbers to tackle gridding artifacts and extract both coarse and fine features from the two parallel paths. The potential impact extends beyond overcoming specific challenges in brain tumor classification, paving the way for greater improvements and developments in related domains.

1.5 THESIS OUTLINE

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, objective and thesis organization. Motivation for the research, objective for the study and how this thesis paper is organized are concisely described in the chapter.
- Theoretical background in line with literature review associated with the study is organized in chapter 2. It will provide a brief overview of the theories, formulas and literature required for the audience to grow interest in and understand the study.
- Chapter 3 systematically organizes the methodological framework followed in the study. A comprehensive flowchart for the methodological framework is illustrated for providing a concise understanding on how the study is performed. Details on designing and modeling of 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 overall study method, results and findings. Challenges and limitations are also discussed. Future recommendations and conclusion finish the thesis paper.

Chapter 2: LITERATURE REVIEW

This chapter offers the reader a thorough theoretical foundation reinforced by a substantial body of literature. In order to define the issue and provide a solution, this chapter will assist with understanding the fundamental vocabulary and concepts of the pertinent subject, such as brain tumors and their detection and classification. Brain tumor terminology and basic concepts are explained in section 2.1. Next, section 2.2 and section 2.3 explain how neural networks, and dilated convolutional neural networks, operate. Section 2.4 describes the principles of classifiers, including the Softmax, Support Vector Machine (SVM), K-Nearest Neighbor (K-NN), Naïve Bayes (NB), and Decision Tree (DT).

2.1 CONCEPT AND TERMINOLOGIES

A tumor is an abnormal mass that can develop on or inside the skull. This unusual and anomalous region of the human brain is referred to by two distinct names: tumor and cancer. Cancer and tumors do not share the same traits. A tumor is an unusual growth of cells, either solid or fluid-filled. Primary and secondary tumors are two types of tumors [15, 16].

The cells of the organ in which the tumor is located make up the primary tumor. The nervous system primarily supports the growth of primary tumors, which grow very slowly. Gliomas are the name for this kind of tumor that is connected to the nervous system. Cancer is the fast and uncontrollable expansion of abnormal cells that harms the brain's surrounding healthy tissues. Benign, Malignant, and Pre-malignant [1, 17] are the three categories for tumors. Non-cancerous symptoms are present in malignant. Pre-malignant tissue possesses precancerous traits. The cells that make up a secondary tumor come from various parts of the body. It spreads swiftly. In another way, it is possible to claim that cancer cells are the root cause of secondary tumors. It follows that while not all

tumors are cancer, all cancers are tumors [18]. Tumors can be categorized using a variety of factors, as listed below:

- Location of tumor in skull
- Location of tumor in brain
- Location of tumor in compartment

The following is a classification of dominant pathology bases:

- Benign
- Malignant

2.1.1 Brain Tumor Diagnosis and Medical Imaging Methods

Early diagnosis facilitates the course of treatment. Various methods, such as brain biopsies and brain imaging systems, are employed to diagnose tumors and determine the cause and consequences of associated diseases.

A brain biopsies involves drilling a hole in the skull, removing a tumor and tissue sample, and using a microscope to examine the tumor's type, composition, and origin. This method poses a serious risk to human life. When doing a biopsy, imaging techniques are also utilized to find the tumor and remove tissue samples [19].

MRIs, CT scans, and x-rays are some of the imaging methods used to obtain images of the brain in order to diagnose tumors based on their location and size.

A CT scan [20] is a crucial imaging method in the medical field that can provide information in just a couple of seconds and typically only takes a small amount of time. Compared to x-rays, it aids in providing clearer information, but there is very little chance of radiation exposure.

Positron emission tomography, or PET, involves injecting a radioactive substance into the bloodstream, which a scanner then detects to produce an image. This method provides information about the functioning and activity of the brain. This technique uses harmful materials at a low cost [21].

X-rays are an imaging method that cannot provide precise information about an origin. If x-rays are applied repeatedly to the same body in the same location, they can result in skin cancer. However, this method is simple to use and less costly [22].

Another method that uses radio frequency signals to create an image of the brain is magnetic resonance imaging (MRI) [23]. The specific method for this study is this form of imaging.

2.1.2 Brain Tumor Characteristics on MRI

Imaging methods such as MRI are more beneficial than x-rays [9]. MR images are free of hazardous radiation and give medical professionals all the information they need to diagnose patients and make decisions. The preprocessing of MR images is utilized in the diagnosis and detection of brain tumors. Depending on the requirements, several MRI types are employed in this procedure. Varieties of MRI sequences, such as T1, T2, and FLAIR that are used as input in the preprocessing stage.

The concepts of TE and TR must be understood in order to comprehend the idea of various MRI image types. The term "time of echo," or TE, refers to the interval of time between an RF pulse being delivered and an echo signal being received. Repetition time, or TR, is the amount of time that passes between two consecutive pulses applied in the same order.

T1-weighted images: has a fluid and CSF appearance that is dark. Compared to white matter, gray matter is darker in color. In brain structure imaging, T1 produces superior results; fat seems brighter in this form of image. Longitudinal relaxation is used to create images with little TE and TR time (TR→500msec, TE→14msec).

T2-weighted images: appear brighter because they have more CSF and fluid signal intensity than tissue. T2 employed a long (TR→4000msec, TE→19msec)

traverse relaxation time for TE and TR to generate images. T2 is better for fluids and water, making it perfect for edematous tissue [24].

FLAIR is similar to T2, but the CSF fluid is lessened, and the abnormalities are still visible. It works well for brain edema imaging. It produces images with very long TE and TR times (TR→9000msec, TE→114msec) [25]. The distinction between these different sequence types in an MR image is shown in Fig. 2.1.

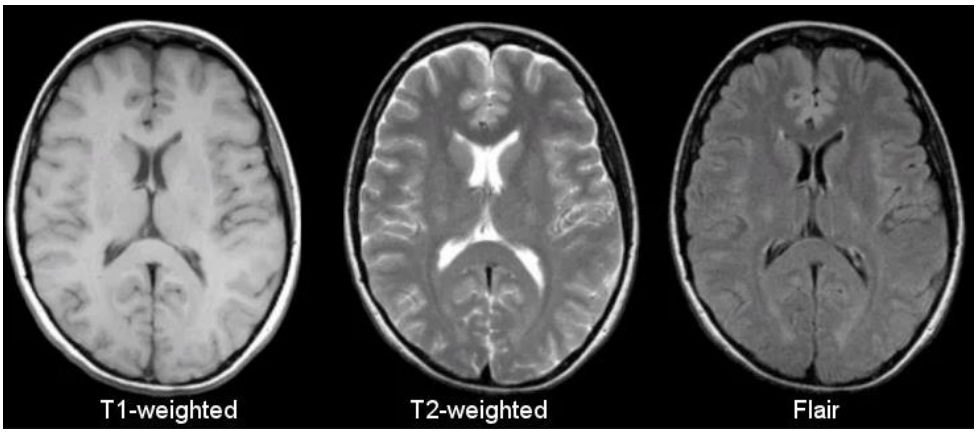


Fig. 2.1 Types of MRI technique [26].

2.2 NEURAL NETWORKS

An artificial intelligence procedure called a neural network teaches computers to process data in a manner that is similar to the way the human brain does. Deep learning is a type of machine learning method that uses neurons or networked nodes arranged in a layer-by-layer structure resembling the human brain [27]. With little assistance from human, computers can make intelligent decisions with the help of neural networks. There are several types of neural networks, including artificial neural network (ANN), convolutional neural network (CNN), and dilated convolutional neural network (CNN). Some important neural networks are listed below.

2.2.1 Artificial neural network

ANNs are a specialized set of algorithms that follow the neuron structure of the human brain and are used in various sorts of classification or pattern recognition tasks. This learning can take a varying amount of time depending on

the complexity of the problem and this is basically how humans learn. ANNs try to mimic this high order learning behavior to find relations in data sets or to model unknown transfer functions. The basic unit of computation in an artificial neural network is called a neuron. All the neurons in a neural network are arranged in layers. A neural network may contain multiples of different types of layers in them. The first or initial layer is called the input layer. This layer contains input neurons and any inputs to the neural network are applied through this layer. As this is the first layer, no processing is done here and the data is just forwarded to the internal layers. The layers between the first or input layer and the last or output layers are called hidden layers. Each hidden layer may hold varying amounts of neurons in them depending on the application. These neurons take weighted inputs from neurons in the previous layer and apply biases to them. After the weighted sums are taken and biases are accounted for, the outputs from each neuron may pass through an activation function and forwarded to the neurons in the succeeding layer. The final layer of a neural network is called the output layer. They also do similar computations as the neurons in the hidden layers [28]. Fig. 2.2 depicts the basic structure of ANN.

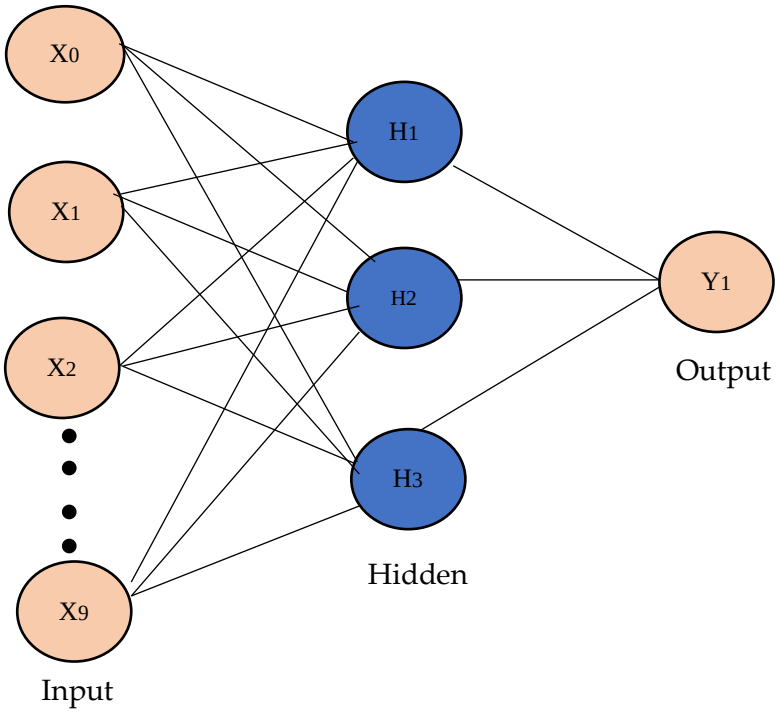


Fig. 2.2 The basic structure of the ANN [29].

2.2.2 Convolutional Neural Network

A deep feed-forward neural network model known as a convolutional neural network is used to analyze grid-like data, such as images [30]. The CNN has excellent feature extraction capabilities due to the construction of many filters, which it employs to extract representative features from input data layer by layer [31, 32]. Fig. 2.3 depicts the basic structure of CNN [33]. The most basic CNN model consists of one trainable feature extraction step and one classification step. The feature extraction step has 3 layers: a convolution layer, an activation layer and a pooling layer. The classification step consists of multiple fully connected layers of a multilayer perceptron [34].

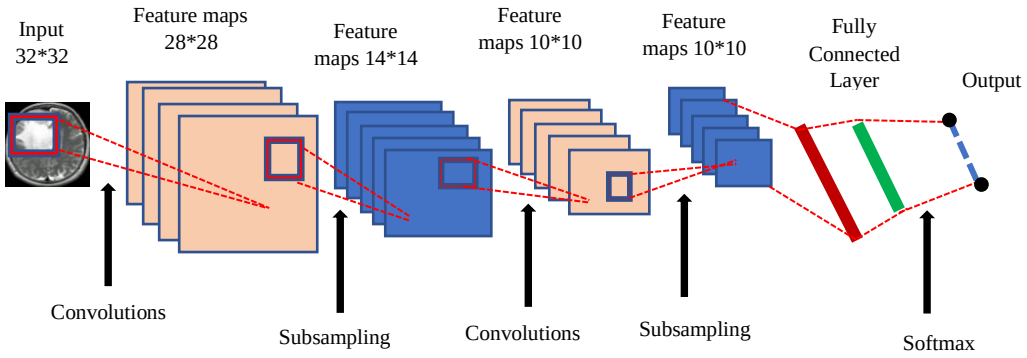


Fig. 2.3 The basic structure of the CNN [33].

2.2.2.1 Convolutional Layer

One of the most significant core modules in CNN is the convolutional layer. It includes a number of convolution kernels. The kernel is often a matrix, with each value representing a weight learned from the input data. An error back propagation method determines these weights automatically [35]. Fig. 2.4 depicts the calculation technique principal schematic diagram of a convolution operation.

Fig. 2.4 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 [35]. The rectangular convolution kernel can be used in the feed-forward calculation process to traverse every element on the whole input feature map in order to obtain the output feature maps of the convolutional layer.

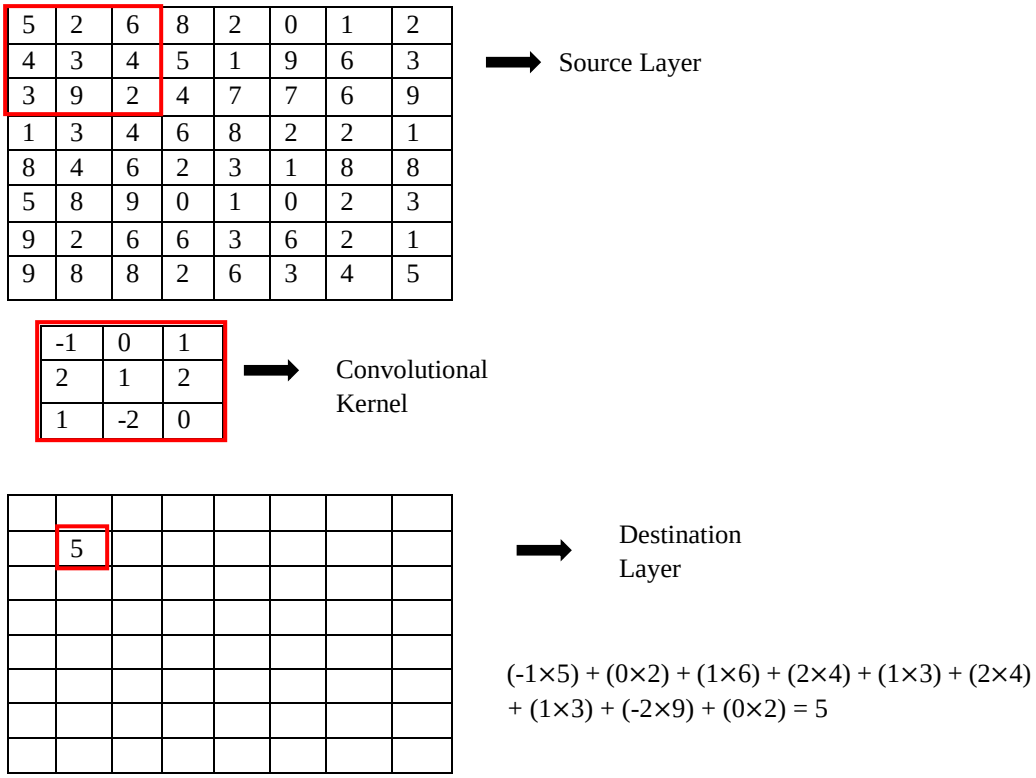


Fig. 2.4 Visualization of the convolutional operation [36].

The convolution operation equation is shown in (2.1).

$$O_{p,q,r}^{l} = f(W_r^l \, I_{p,q}^{l-1} \; + \; b_r^l) \tag{2.1}$$

Where for r^{th} kernel in layer l, $O_{p,q,r}^{l}$ expresses the resultant feature map of position (p, q), W_r^l represents the weight vector's values, $I_{p,q}^{l-1}$ indicates the input vector of position (p, q) in the $l-1$, and b_r^l is the symbol of bias. In addition, the activation function is $f(.)$ [37].

2.2.2.2 Activation Functions

Following the convolution operation, the output of the convolution layer is subjected to a nonlinear transformation activation function. Nonlinear transformations can be obtained using various activation functions.

The most popular activation parameter for FC layers is the Rectified Linear Unit (ReLU), which is illustrated in (2.2) [38].

$$\boldsymbol{f(x)} = \begin{cases} x & for\ x \geq 0 \\ 0 & for\ x < 0 \end{cases} \tag{2.2}$$

It should be mentioned that the final FC layer's activation function is usually SoftMax for the categorization of multiple classes and Sigmoid for binary classification. The node values in the final FC layer of a CNN can be computed using (2.3), and the sigmoid activation function for a binary categorization issue is illustrated in (2.4) [37,39].

$$z = w^T h + b \tag{2.3}$$

$$P(y = 1|x) = \frac{1}{1+\exp(-z)} \tag{2.4}$$

Where, h stands for the NN layers' internal calculations, b shows the bias, and w stands for the weights used to determine an output node's value. Furthermore, the input vector and output class are denoted by x and y, respectively. The SoftMax activation function is shown in (2.5) for the multi-class categorization problem.

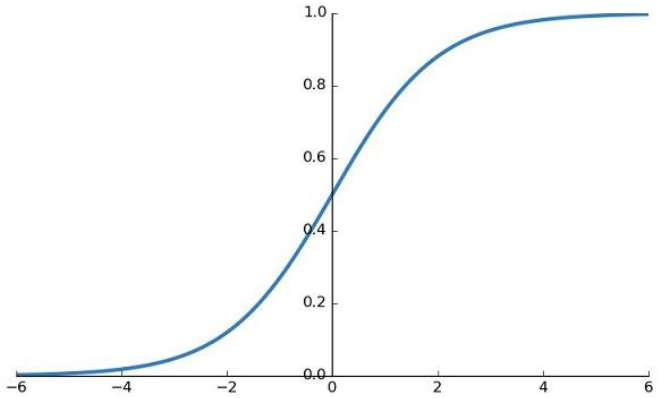
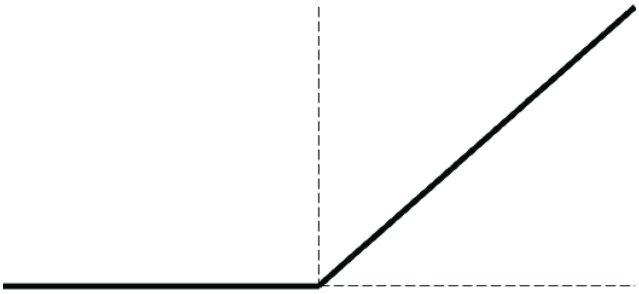
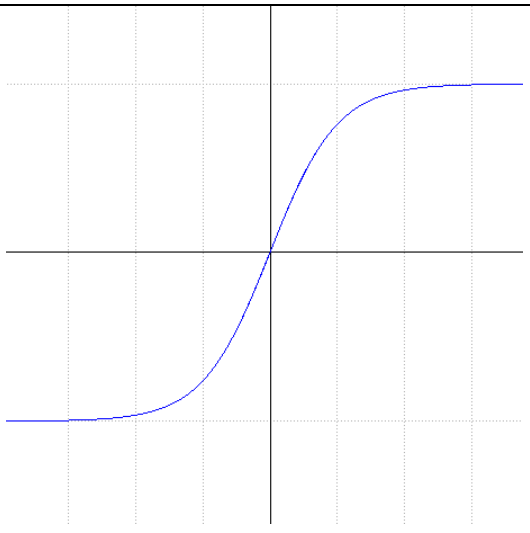
$$P(y|x) = \frac{\exp(f_y)}{\sum_{c=1}^{C} \exp(f_c)} \tag{2.5}$$

where, x stands for the input vector and y for the class in the case of a multi-class categorization problem. Additionally, the c^{th} component of the class rating vector in the final FC layer is displayed by f_c. The category k with the highest P coefficient is chosen as the output class in the SoftMax activation function [40].

The Sigmoid, hyperbolic tangent, and ReLU are the most often utilized activation functions in CNN. The sigmoid's input value x is −∞ ~ +∞, and the sigmoid's output value f (x) is 0–1. The output value f (x) of tanh functions is –1–1. Although both of these functions are better at nonlinear transformation, they also have an issue with gradient fading and saturation. The output change values f (x) of the two functions practically equal zero when the input value x's absolute value is high. To address this issue, the ReLU activation function only evaluates the forward signal. The ReLU activation function only considers the forward signal, eliminating the influence of the negative signal, and it has an excellent

fitting ability and sparsity, which considerably improves calculation efficiency [39]. Table 2.1 shows the equations and graphs.

Table 2.1: Activation Functions in a CNN

Name of the function	Expressions	Graphs
Sigmoid	$\int(x) = \frac{1}{1+e^{-x}}$	
Relu	$\int(x) = \begin{cases} x, x \geq 0 \\ 0, x < 0 \end{cases}$	
Tanh	$\int(x) = \frac{1-e^{-2x}}{1+e^{-2x}}$	

2.2.2.3 Pooling Layer

The primary goals of pooling are to reduce the dimension and quality of trainable parameters of a CNN and to separate the most significant representative

features from the activation layer's output feature maps. For feature screening, a rectangular pooling kernel is primarily used in the pooling operation.

Fig. 2.5 shows the calculating process, principle schematic diagram, and pooling operation.

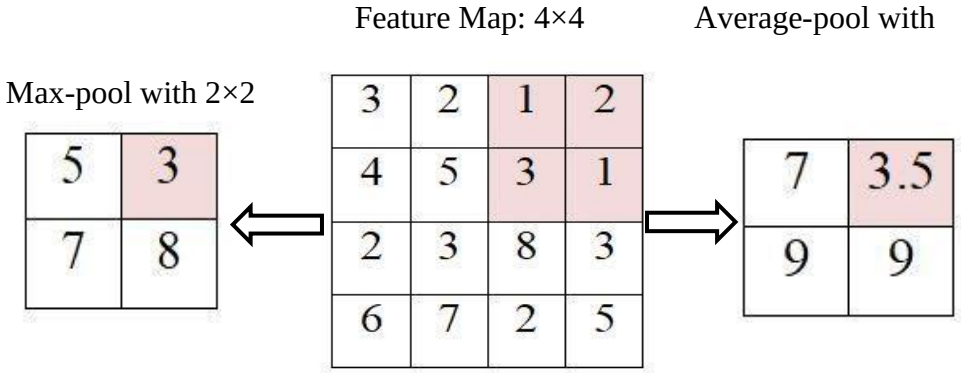


Fig. 2.5 Visualization of the pooling operation.

The two main types of pooling operations are maximum pooling and average pooling. The max-pooling method chooses the highest value possible from each element in the relevant area of the pooling window to serve as its representative value.

In the relevant area of the pooling window, average pooling determines the average value from all the elements as the representative value. Average pooling is more sensitive to background information, while max-pooling excels at managing texture features [41].

By down-sampling, pooling layers lower the dimensionality of the feature maps. The stride, padding, and filter size are among the hyper-parameters that comprise pooling layers, although they do not contain any other parameters. Two common varieties of pooling layers are max pooling and global average pooling. Patches are taken from the feature maps and the greatest value in each patch is chosen as the output in the max pooling process. By averaging over all of the feature map's elements, global average pooling turns a feature map into a 1x1 array. (2.6) is used to calculate the output size following a convolution/pooling operation in a CNN [40].

$$O = \left[\frac{n-f+2p}{s}\right] + 1 \tag{2.6}$$

Where, n stands for the dimension of input, f is the kernel size, the padding size is shown by p, and s is symbol of stride size.

2.2.2.4 Fully Connected Layer

The pooling layers' feature maps are smoothed out and sent to several one-dimensional (1D) vectors known as FC Layer or dense layers. The number of nodes in the final FC layer corresponds to the number of classes in a given classification task. FC layers' hyper-parameters, which comprise the number of weights and activation functions.

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. A CNN's flatten function converts the final output feature map into a one-dimensional array. The primary responsibility is to further extract the features from the output data of CNN and to link the feature extraction stages with the SoftMax classifier. All neurons between layers are interconnected in a network that is fully connected, as described below [41]:

$$O\ (X) = \int f\ (W\ \times X + B) \tag{2.7}$$

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) [41].

2.2.2.5 Cross Entropy Loss Function

A backpropagation algorithm is used during CNN training to adjust the weights of the FC and convolution layers. The two main elements of backpropagation are the loss function and Gradient Descent (GD), in which GD is used to minimize the loss function. Among the loss functions most frequently employed by CNNs is the Cross-Entropy (CE) loss function. Consequently, it is possible to compute the CE loss function for a binary categorization issue with sigmoid activation function, as shown in (2.8) [39].

$$L = \frac{1}{N} \sum_{i=1}^{N} - [y_i\ log\left(\frac{1}{1+exp(-z)}\right) + (1 - y_i)\ log(\frac{exp(-z)}{1+exp(-z)})] \tag{2.8}$$

Where, $z\ is$ computed using formula (2.3). The CE loss function can be expressed as (2.9) for a multi-class categorization issue with the SoftMax activation function [39].

$$L = \frac{1}{N}\sum_{i=1}^{N} - log(\frac{exp(f_{yi})}{\sum_{c=1}^{C} exp(f_c)})] \quad (2.9)$$

Where, N denotes the quantity of training elements, input image class i^{th} is indicated by y_i, and the c^{th} component of the category scores vector in the final FC layer is presented by f_c [39].

GD is a method for limiting an $theta - parametrized$ objective function $J(\theta)$. Batch GD, Stochastic GD (SGD), and mini-batch GD are the three variations of GD. Employing (2.10), each variable in the dataset gets modified for every training sample in batch GD. SGD updates the parameters for every training sample in the dataset, unlike batch GD. A mini-batch of n samples from the dataset is updated for variables in mini-batch GD. SGD and mini-batch GD formulas are shown in (2.11) and (2.12), accordingly. The variables in these equations are the capable of learning variable, θ, the objective function $J(\theta)$ that needs to be lowered, the learning rate η, and $\nabla_\theta J(\theta)$ indicates the gradient of the objective function regarding parameters θ [39].

$$\theta := \theta - \eta.\nabla_\theta J(\theta) \quad (2.10)$$

$$\theta := \theta - \eta.\nabla_\theta J(\theta;\ x^{(i)}; y^{(i)}) \quad (2.11)$$

$$\theta := \theta - \eta.\nabla_\theta J(\theta;\ x^{(i:i+n)}; y^{(i:i+n)}) \quad (2.12)$$

2.3 DILATED CONVOLUTIONAL NEURAL NETWORK

Expanding the receptive field in deep learning involves boosting the dimension and depth of the convolution kernel, which in turn enhances the number of elements in the network. By adding weights of zero to the conventional convolution kernel, dilated convolution may enhance the receptive field without adding more network elements. A 1-dilated convolution kernel similar to the conventional 3×3 convolution kernel is depicted in Fig. 2.6 (a). A 2-

dilated convolution kernel with a 5×5 receptive field is shown in Fig. 2.6 (b); the remaining weights have been assigned zero and nine acceptable weights remain.

1	2	3
4	5	6
7	8	9

(a)

1	0	2	0	3
0	0	0	0	0
4	0	5	0	6
0	0	0	0	0
7	0	8	0	9

(b)

Fig.2.6 Parallelism of conventional 3-by-3 convolution kernels with 5-by-5 2-dilated convolution kernels.

Equation (2.13) defines the convolution operator * as follows: 1-D dilated convolution using dilation rate l = 1 connects input picture F alongside kernel k. The term "standard convolutional neural network" refers to the 1-D convolution. The network is identified as dilated convolutional neural network when the dilation rate l rises.

$$(F * k)(p) = \sum_{s+t=p} F(s) \times k(t) \tag{2.13}$$

Upon the introduction of a dilation rate when a dilation factor called l is introduced and by generalizing this factor l that can be defined as,

$$(F *_l k)(p) = \sum_{s+lt=p} F(s) \times k(t) \tag{2.14}$$

A fundamental convolution neural network has a value of $l = 1$[42].

2.4 CLASSIFIERS

A type of machine learning algorithm called a classifier is employed in data science to categorize on input of data. Labeled data is utilized to train classifier algorithms; for example, in image identification the classifier receives the training data with labels for the images. After obtaining sufficient training, the classifier can be trained to accept unlabeled images and produce categorization labels for each image. Classifier algorithms use sophisticated mathematical and statistical techniques to predict the probability that a given data input will be classified in a specific way [43]. Data learning uses a variety of classifier types, including Softmax, K-Nearest Neighbor (K-NN), Decision Tree (DT), Support Vector Machine (SVM), and others.

2.4.1 SoftMax Classifier

SoftMax classifier is a supervised learning method that is typically used when there are numerous classes involved, in contrast to logistic regression classifier which is used for binary class classification. Each class in the SoftMax classifier is given a probability distribution. All other probabilities are scaled in accordance with the probability distribution of the class that has the highest likelihood, which is normalized to 1. The output of the neurons is similarly converted into a probability distribution over the classes via a softmax function. It possesses the following qualities [43]:

- It has characteristics with the logistic sigmoid, which is also linked and used in probabilistic modeling.
- It accepts values in the range of 0 and 1, where 0 represents impossibility and 1 represents a certainty.
- Given an input x, the softmax derivative with respect to that input can be seen as a prediction of the likelihood that a specific class would be chosen.

2.4.2 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 SVM algorithm is a categorization technique that works by creating a hyperplane with the maximum margin between classes.

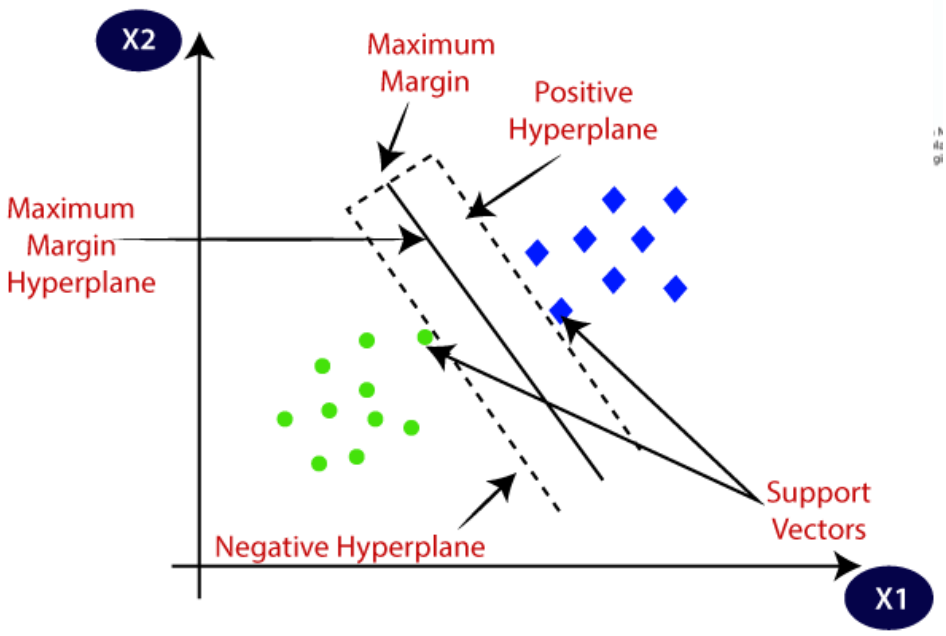


Fig.2.7 Visualization of the SVM classification process [44].

An SVM needs a collection of instances, each of which can be expressed as a pair of (x_i, y_i). x_i is an input vector's symbol and its matching class label is indicated b y_1. Taking into consideration a training set of S = $[(x_1, y_1), \ldots, (x_l, y_l)]$, in which $x_i \in \mathbb{R}^d$ and $y_i \in [+1, -1]$. The two categories' splitting hyperplanes are computed using (2.15) [45].

$$y_i\,(w^T x_i + b) \geq 1 - \xi_i\,, \quad \forall_i = 1 \ldots\ldots\ldots\ldots l \tag{2.15}$$

Where, in case of data points, ξ_i represents non-negative stack variables. The concluding SVM optimization problem is given as (2.16) to (2.18) applying Lagrange dual [45].

$$\text{Max-}\ {}^1\!/_2 \sum_{i,j=1}^{l} \alpha_i\,\alpha_j\,y_i\,y_j\ \ K\left(x_i, x_j\right) + \ \sum_{i=1}^{l} \alpha_i \tag{2.16}$$

$$\sum_{i=1}^{l} \alpha_i\,y_i = 0 \tag{2.17}$$

$$0 \leq\ \alpha_i\ \leq C, \quad \forall_i = 1 \ldots\ldots\ldots\ldots l \tag{2.18}$$

Where, $K\ (x_i, x_j)$ represents kernel function, $\propto_i$ denotes the multipliers of Lagrange, and parameter C ≤ ∞ is a penalty. Particularly,$\alpha_i \neq 0$ exclusively for support vectors with x_i. As shown in (2.19), an actually separable case is handled using a linear kernel.

$$\text{Linear: } K\ (x_i, x_j) = x_i^T\ x_j \tag{2.19}$$

Both the Kuhn-Tucker condition and the optional weight vector are shown in (2.20) and (2.21) [45].

$$w^* = \sum_{i=1}^{l} \alpha_i^*\ y_i x_i \tag{2.20}$$

$$\alpha_i^* \left[\ y_i\left(w^{*^T} x_i + \ b^*\right) - 1 + \xi_i\ \right] = 0 \tag{2.21}$$

In SVM, the ideal hyperplane is computed as illustrated in (2.22) [45].

$$F(x) = \sum_{i=1}^{l} \alpha_i^*\ y_i K\ (x_i, x) + \ b^* \tag{2.22}$$

Lastly, the definition of the decision function is (2.23) [45]

$$C(x) = sign\ (\sum_{i=1}^{l} \alpha_i^*\ y_i K\ (x_i, x) + \ b^*) \tag{2.23}$$

For non-linearly distinguishable categories in SVM, two of the most popular kernel operations are polynomial and RBF kernels. The expressions (2.24) and (2.25) illustrate the formulas for both of those kernel functions, where r and γ stand for the hyper-parameters of the polynomial and RBF kernels, accordingly [45, 46].

$$\text{Polynomial: } K\left(x_i, x_j\right) = \ (x_i^T\ x_j\ + 1)^r,\ \mathrm{r} \in \mathbb{Z}^+ \tag{2.24}$$

$$\alpha_i^* \left[\ y_i\left(w^{*^T} x_i + \ b^*\right) - 1 + \xi_i\ \right] = 0 \tag{2.25}$$

2.4.3 KNN Classifier

Because of its simplicity and efficiency, k-nearest neighbor classifier is a popular classifier. It has been referred to as a lazy learner because its training phase consists 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. A machine learning classification algorithm called KNN determines the distance between each training set of data and a given sample in order to complete the classification task. Based on the majority vote on the labels of the k

nearest neighbors that were chosen, a label is thus assigned to the given sample [47].

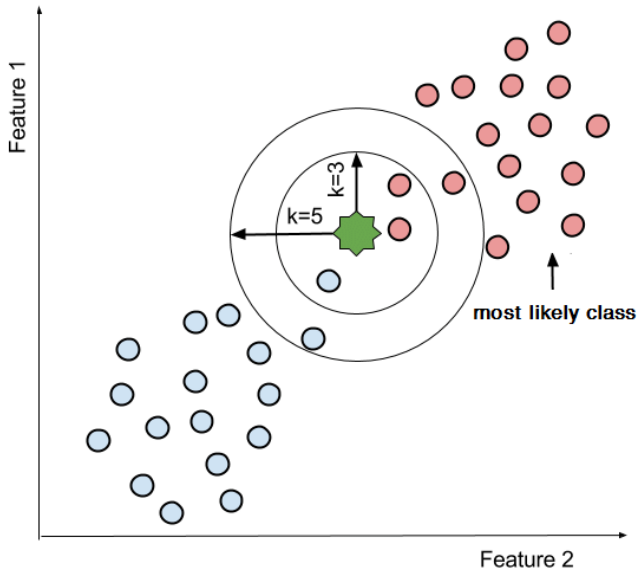


Fig.2.8 KNN classification for a two-class problem when the k parameter is set to "3"and "5" (the "pink" and the "blue" circles). An invisible sample point is indicated by the green star [47].

With x' serving as the data point and y' as its corresponding class, a training set D and a test object z = (x', y') are considered. The KNN algorithm calculates the separation between z and each of the points (x, y) ∈ D, where x represents training set data and y indicates class-corresponding data. The test object is given the majority category among its nearest neighbors based on (2.26) after acquiring the K nearest neighbors' collection D_z [48].

$$y' = \arg max_v \sum_{(x,y)\in D_z} I(v = y_i) \tag{2.26}$$

Where, v stands for a class, y_i represents the class of i^{th} closest neighbor, and I (·) for an indicator function, which is equal to 1 in the case that its statement is true and equal to 0 in the other case. The value of the results is greatly affected by the choice of K. For instance, if the coefficient K is too large, lots of data points from different classes will be included in the list of nearest neighbors, and a very small K value may render the result sensitive to noise points. Another crucial

component in creating a KNN technique is the distance measuring. A method for distance calculation used in the construction of a KNN classifier is Euclidean distance. In (2.27), the Euclidean distance function is illustrated [48].

$$d\,(m, n) = \sqrt{(m_1 - n_1)^2 + \cdots + (m_p - n_p)^2} \tag{2.27}$$

Where, $m = (m_1, m_2, \ldots, m_p)$ and $n = (n_1, n_2, \ldots, n_p)$ denote two data points having p characteristics.

2.4.4 Naïve Bayes Classifier

A machine learning classification technique called Naïve Bayes (NB) predicts the final class using vectors of attributes and class prior probabilities, as shown in (2.28) [48].

$$P\,(C = i \mid X = x) = \frac{P\,(X=x \mid C=i) * P(C=i)}{P\,(X=x)} \tag{2.28}$$

Where, $x = (x_1, x_1, \ldots, x_n)$, indicates the set of all attributes taken form a region $\Omega = D_1 * \ldots * D_n$ and C is the resultant class representative. $P\,(X = x \mid C = i)$ in the NB classifier is computed by taking the dataset's features to be independent, and calculating the probability as shown in (2.29) [48].

$$P\,(X = x|C = i\ ) = \prod_{j=1}^{n} P\left(x_j \,\middle|\, C = i\right) \tag{2.29}$$

In the NB approach, the category that has the highest probability is chosen to be the expected class of an input vector. Each group in the NB classifier has the same term $P\,(X = x)$. As such, it can be disregarded, and the following is the final classification formula [48, 49]:

$$C^*\,(x) = \arg\, max_i\, P(X = x \mid C = i) \times P(C = i) \tag{2.30}$$

2.4.5 Decision Tree Classifier

Although decision trees, a supervised learning technique, are generally preferred for classification, they can be utilized for solving regression and classification problems as well. It is a classifier based on a tree structure, where each leaf node represents the classification outcome, internal nodes represent dataset features, and branch represent decision rules. Two nodes make up a decision tree: Decision node and Leaf node. Decision nodes are used to make

decisions and have many branches, whereas leaf nodes are the outcomes of decisions and do not have any more branches. The general layout of a decision tree is shown in Fig. 2.9.

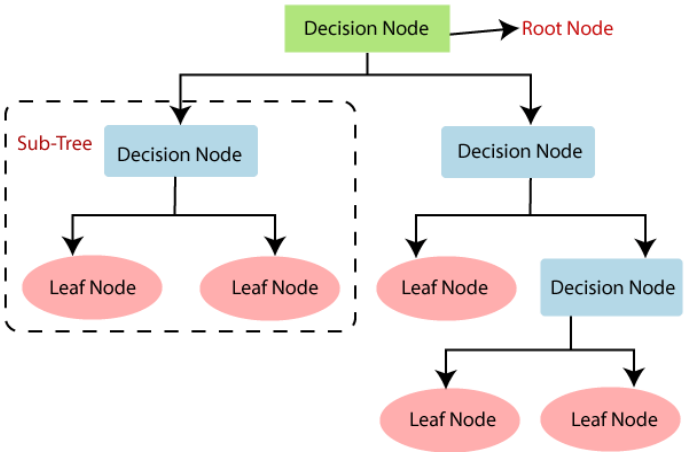


Fig.2.9 The general structure of the decision tree [50].

The Decision tree structures represent an approach to non-parametric supervised learning that can be employed for classification as well as regression. The goal is to create a model that predicts the value of an objective variable using fundamental decision rules inferred from the data features. Decision trees have a number of benefits.

- Minimal data preparation is needed. For the purposes of data normalization, dummy variable creation, and blank value removal, other techniques are commonly applied.
- Efficient in comprehension and interpretation. One can envision trees.
- The logarithmic cost of data prediction increases with the quantity of data points utilized for training the structure of the tree.
- Capable of managing data that is both a categorical and numerical. The other approaches are typically focused on examining datasets with a single kind of parameter.
- Capable of managing multiple output issues.

With training vectors $x_i \in \mathbb{R}^n$, i= 1,…, l and a label vector $y \in \mathbb{R}^l$, a decision tree successively divides the domain of features so that instances that share similar desired values are organized together [48].

Let the data at node m be denoted by Q_m having n_m instances. Splitting $\theta = (j, t_m)$ for every candidate which is composed of a characteristic j, and threshold t_m, dividing the data into $Q_m^{left}(\theta)$ and $Q_m^{right}(\theta)$ subgroups [48]

$$Q_m^{left}(\theta) = \{(x, y) | \, x_j \leq \, t_m\} \tag{2.31}$$

$$Q_m^{right}(\theta) = Q_m \setminus \, Q_m^{left}(\theta) \tag{2.32}$$

Next, a loss function H () is used to calculate the quality of a potential split of node m [48].

$$G(Q_m, \theta) = \frac{n_m^{left}}{n_m} H\left(Q_m^{left}(\theta)\right) + \frac{n_m^{right}}{n_m} H(Q_m^{right}(\theta)) \tag{2.33}$$

To determine the settings that will reduce the impurity [48].

$$\theta^* = argmin_\theta \, G \, (Q_m, \theta) \tag{2.34}$$

Carry out for $Q_m^{left}(\theta^*)$ and $Q_m^{right}(\theta^*)$ subsets until the maximum permitted depth is achieved, $n_m < \, min_{samples}$ or $n_m = 1$. [48]

2.4.6 Average Ensemble Method

Machine learning and signal analysis both use the statistical technique of ensemble averaging. Model averaging is a machine learning technique for ensemble learning in which each member of the ensemble makes an equal contribution to the ultimate prediction. A group of frameworks often outperforms a single one because the individual errors in the models "average out." Two characteristics of artificial neural networks are the foundation of the ensemble averaging concept.

- It is possible to decrease bias in any network, but doing so can boost variance.
- It is possible to minimize the variance in an ensemble of networks without affecting bias.

Through ensemble averaging, a set of low-bias, high-variance networks are combined to produce a new, low-bias, low-variance structure. The theory provides a clear method for assembling a group of experts with high variance and low bias, then averaging them. This basically means that a set of experts with different parameters are created; these are often the initial synaptic weights, but other parameters (like the learning rate, momentum, etc.) can also be varied. Certain authors advise against stopping too soon and varying weight decay. As a result, the actions are:

- Create N experts, each starting at a different value: Typically, initial values are selected at random from a distribution.
- Train every specialist independently.
- Add up all the experts and take the mean of their scores.

A more sophisticated form of the ensemble average sees the outcome as a weighted total rather than just the average of all the experts. Assuming that every expert is y_i, then the total outcome $\bar{y}$ can be described as follows:

$$\bar{y}(X, \alpha) = \sum_{J-1}^{P} \alpha_j \, y_j \, (X) \quad \text{where, } \alpha = \text{weighted set} \tag{2.35}$$

Neural networks are used to solve the optimizing issue of identifying alpha [51, 52].

Theoretical background in line with literature review associated with the study is observed in this chapter. It provides a brief overview of the concept and terminologies of brain tumor diagnosis and medical imaging methods, brain tumor characteristics on MRI, different types of neural networks, and machine learning classifiers that are used in this research, required for the audience to grow interest in and understand the study.

Chapter 3: MATERIALS AND METHODOLOGY

The materials and methodology used in the investigation are covered in detail in this chapter. The suggested approach is described in Section 3.1. The details of the experimental data are described in Section 3.2.The data from the experiment and its pre-processing methods are described in Section 3.3. The methods for enhancing experimental data are described in Section 3.4. Next, using the average ensemble method, the developed dilated PDCNN is explained in Section 3.5. In Section 3.6, the machine learning classifiers- SVM, KNN, NB, and Decision Tree techniques- are explained.

3.1 OVERVIEW OF METHODOLOGY

The biggest issue before starting treatment is identifying and categorizing tumors from brain MRI scans. There aren't many studies on tumor diagnosis as a time-saving method, despite the majority of brain tumor identification research focusing on tumor slicing and positioning methods. Most DCNN-based methods are unable to acquire global context details of larger regions because of the small receptive fields. Stacking multiple dilated convolutions has the disadvantage of creating a grid effect, even though dilated convolution maintains data resolution at the output layer and expands the receptive field without incorporating calculation. In the event that the dilation factor is low, the model may have a smaller receptive field but misses the coarse characteristics. In contrast, when the dilation factor is excessive, the model is unable to learn from the finer details. This study proposes the use of a dilated PDCNN architecture that maintains a large receptive field to cope with gridding distortions and capture both coarse and fine attributes from images. Initial input image resizing is followed by grayscale transformation to minimize complexity. Data augmentation has since been used to expand the number of datasets. While maintaining an extensive receptive field, dilated PDCNN utilizes the reduced computational cost and helps

to reduce gridding artifacts. The schematic representation of the suggested dilated PDCNN design is presented in Fig. 3.1.

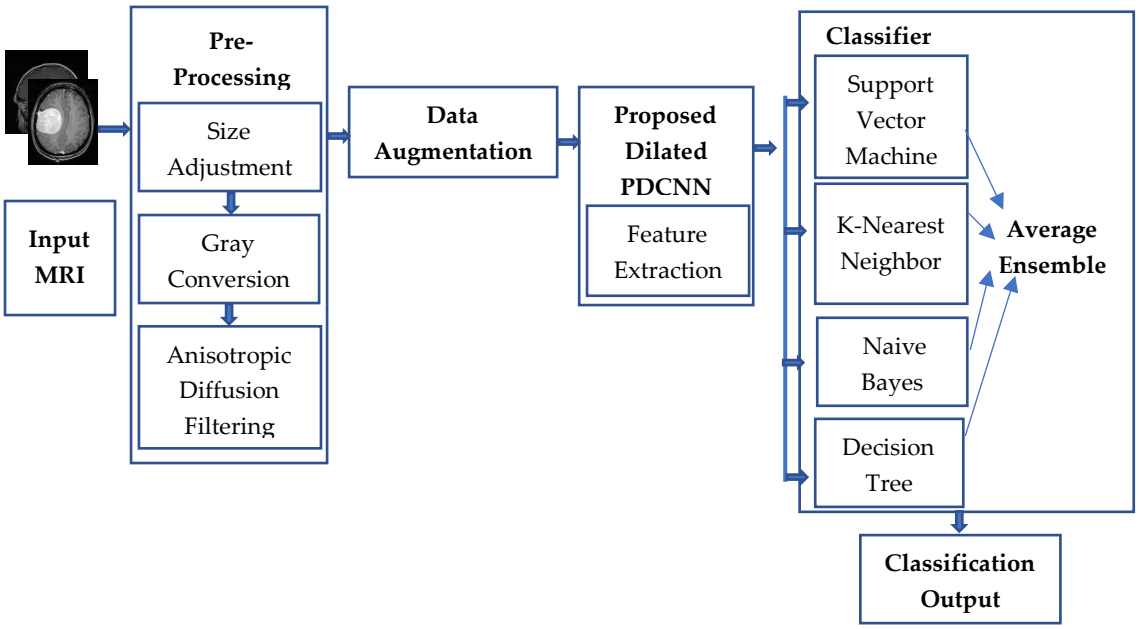


Fig.3.1 Proposed methodology's workflow.

The sequence that follows is the order in which the recommended structure's events occur: brain MRI images are fed into the input layer of the dilated PDCNNs after being processed. The initial images are converted from various resolution dimensions to 32 × 32 pixels for training reasons. The grayscale transformation of these input images contributes to a reduction in complexity. Following that, new images are created from prior ones using data augmentation. The data set has been split into training and testing subsets in order to train the suggested network. The PDCNN structure then makes use of the chosen dilated rates to effectively classify the input images. Following the classification of the images using four classifiers (SVM, KNN, Decision Tree, and Naïve Bayes), the brain tumor identification process is completed using an average ensemble approach.

The step-by-step flow of the suggested framework is mentioned in Algorithm 1.

Algorithm 1 Algorithm Based on Brain Tumor Detection and Classification Approach

Input Representative three different public datasets of brain MRI pictures: $\boldsymbol{I_j, j = 1, 2, \ldots, K}$ of size $\boldsymbol{W \times H}$ and the class labels $\boldsymbol{C = 2, 3, 4}$.

Output Brain Tumor Detection & Classification

1:	Develop a Dilated PDCNN with selected parameters
2:	Accuracy ⟵ 0
3:	**for** epoch = 1, 2, ….. , $\mathbf{N}_{epochs}$ **do**
4:	**for** image = 1, 2, ……, $\mathbf{K}_{batch\_size}$ **do**
5:	$Image_{resized}$ ⟵ Resize (Image, Set $\boldsymbol{W = 32, H = 32}$)
6:	$Image_{gray}$ ⟵ Grayscale, Using $Image_{gray} = rgb2gray(Image_{resized})$
7:	$Image_{data}$ ⟵ Augmentation ($Image_{gray}$)
8:	Input layer takes $Image_{data}$ and sends it to the convolution layers to extract features
9:	Suggested ***Parallel Dilated Deep Convolutional Layers***
10:	Train the dilated PDCNN model with ML classifiers including SVM, KNN, NB, and Decision Tree.
11:	Calculate the test accuracy for each ML classifier with the average ensemble
12:	Calculate error rate, e(t)
13.	**end for**
14.	**end for**

3.2 DATASET DESCRIPTION

This study makes use of three distinct public datasets containing images from brain MRIs. The details regarding the dataset are provided as follows.

3.2.1 Dataset-I

Through the Kaggle platform, the initial accessible dataset of binary-class MRI scans of the brain has been obtained for simplicity and this dataset is widely used. This data is known as dataset-I in this study [8]. This set of 253 brain MRI images includes 98 samples with tumors and 155 samples without tumors.

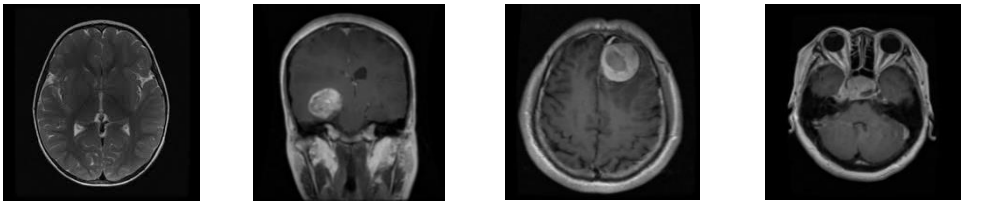
3.2.2 Dataset-II

The Figshare dataset containing 233 patients' brain MRI images is employed in this research [13]. These brain MRI images are obtained at Nanfang Hospital and General Hospital, two Chinese medical centers. This dataset, designated dataset-II, comprises 3064 brain MRI scans, including 1426 glioma tumors, 708 meningioma tumors, and 930 pituitary tumors.

3.2.3 Dataset-III

The additional dataset utilized in this study can also be obtained via the Kaggle website [14]; it contains brain MRI images of glioma tumor, meningioma

tumor, no tumor, and pituitary tumor, numbered 826, 822, 395, and 827, in that order. This collection of data is identified as dataset-III in the current research. The four different kinds of brain MRI images that are present in dataset-III are shown in Fig. 3.2.



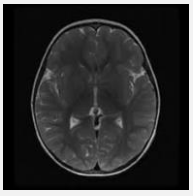
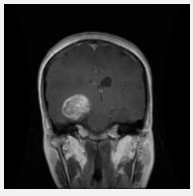
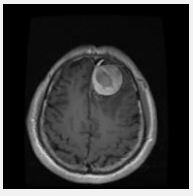
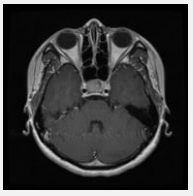
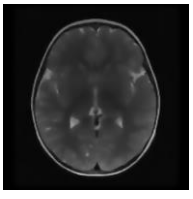
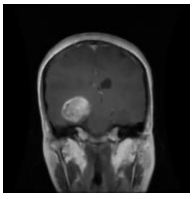
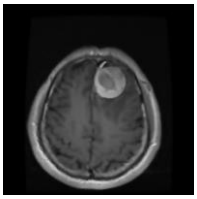
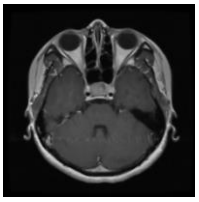
(a) No Tumor (b) Glioma Tumor (c) Meningioma Tumor (d) Pituitary Tumor

Fig. 3.2 Examples of brain MR images [14].

3.3 PREPROCESSING

A method for enhancing the efficiency of a machine learning model is called data preprocessing, which involves purifying and preparing data for usage by the model. The skull photos in the MRI datasets are not all identical in width, and height; instead, each image is scaled to 32 x 32 pixels for training purposes. Grayscale conversion of these data contributes to a reduction in the level of complexity. Digital images can be noise-free without having their edges blurred through the utilization of the anisotropic diffusion filter.

Table 3.1. Filtered Dataset after Utilization of Anisotropic Diffusion Filter.

	No Tumor	Glioma Tumor	Meningioma Tumor	Pituitary Tumor
MRI Brain Pictures [12]	Input image 	Input image 	Input image 	Input image 
Anisotropic Diffusion Filtered Pictures	Filtered image 	Filtered image 	Filtered image 	Filtered image 

3.4 DATA AUGMENTATION

Since deep learning needs a lot of data to extract information, data enhancement is being employed at this time to increase the quantity of available data by altering the initial image. Supplementary data can be used to increase the effectiveness of categorized outcomes. Illustrations can undergo the following procedures: shifting, scaling, translation, and filtering methods. This article uses the process of anisotropic diffusion filtering as augmentation. It can efficiently retain sharp transitions and smooth out regions with uniform intensity, producing clear pictures with improved features.

Table 3.2. Dataset Statistics.

Category			Original Data		Augmented Data	
			Number	Percentage	Number	Percentage
Dataset I	Tumor	Yes	98	61%	196	61%
		No	155	39%	310	39%
	Total		**253**	**100%**	**506**	**100%**
Dataset II	Glioma		1426	47%	2852	47%
	Meningioma		708	23%	1416	23%
	Pituitary		930	30%	1860	30%
	Total		**3064**	**100%**	**6128**	**100%**
Dataset III	Glioma		826	28.78%	1652	28.78%
	Meningioma		822	28.64%	1644	28.64%
	Pituitary		395	13.76%	790	13.76%
	No Tumor		827	28.81%	1654	28.81%
	Total		**2870**	**100%**	**5740**	**100%**

3.5 DEVELOPED DILATED PDCNN DESIGN

This research presents the design of a multiscale dilated two simultaneous deep CNN technique to extract multiscale detail characteristics from MRI images. To increase the receptive field despite adding more parameters to the network, dilated convolution is used. Additionally, batch normalization is used to guarantee that the model's precision won't drop as the network depth increases.

The multiscale extraction of characteristics, integrating path, and classification stage are the three main elements of the suggested network, as

illustrated in Fig. 3.3. Since the suggested model uses dilated CNNs, the DF is an additional hyper-parameter that must be considered.

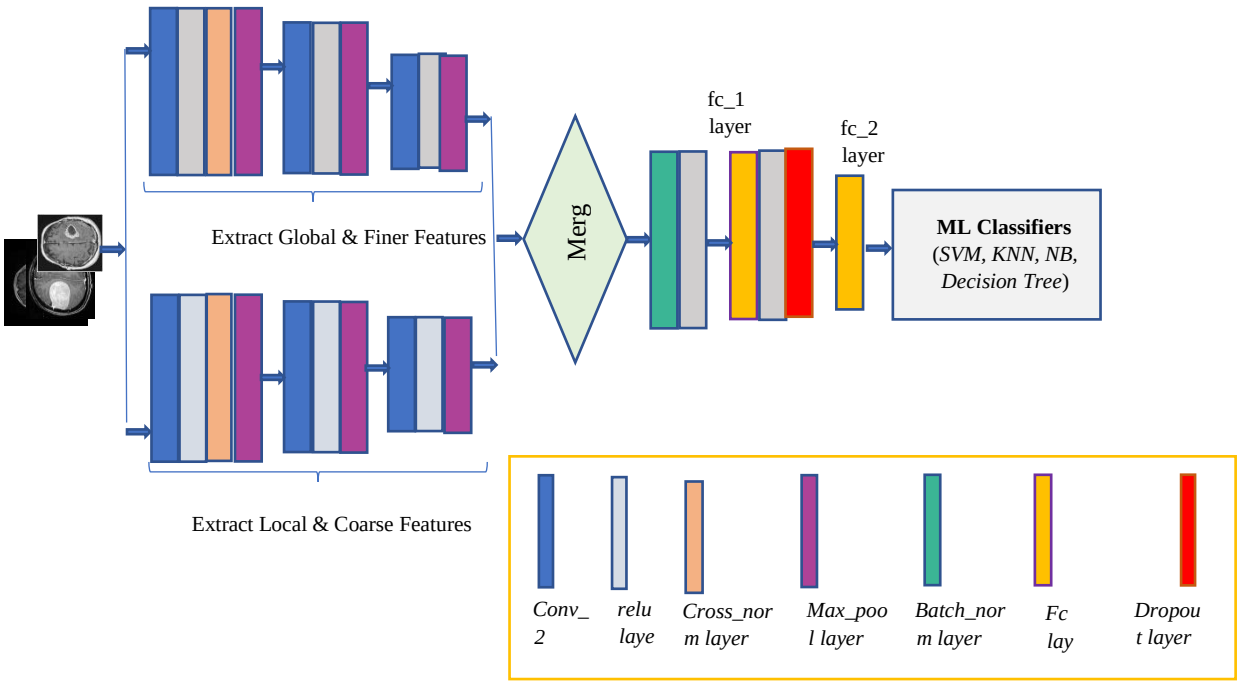


Fig. 3.3 Proposed architecture of dilated PDCNN model.

Both local and global characteristics are acquired in the dilated PDCNN framework through the corresponding local and global routes. However, most DCNN-based methods cannot effectively collect both local and global data because of their tiny receptive fields. Stacking multiple dilated convolutions has the disadvantage of creating a grid effect, even though dilated convolution maintains data resolution at the output layer and expands the receptive field without incorporating computation. In the event that, with poor DF, the model may contain a smaller receptive field nevertheless misses the coarse features. In contrast, with the excessive DF, the model is unable to pick up from the finer details. By contrasting various DFs, these suitable DFs are chosen for both local and global feature paths. Each of the convolutional layers is followed by the max-pooling layer for every single path that down samples the outcome of the convolutional layer and uses the ReLU activation function. In the end, an average ensemble method is employed to carry out the brain tumor categorization process after four ML classifiers—have been training the images.

The step-by-step flow of the suggested dilated PDCNN structure is mentioned in Algorithm 2.

Algorithm 2 Algorithm Based on Dilated PDCNN Model

	Parallel Dilated Deep Convolutional Layers
i.	For the local and coarse path set the window size = 5 and for the global and finer path set the window size = 12
ii.	Divide $\boldsymbol{I_j, j = 1, 2, \dots, K}$ into blocks $\boldsymbol{a_{ij}, i = 1, 2, 3, \dots, B, j = 1, 2, 3, \dots, K}$, of size $\boldsymbol{w \times h \times d, where\ d}$ = Number of feature maps in $\boldsymbol{I_j}$ and $\boldsymbol{B}$ = Number of blocks created from each $\boldsymbol{I_j}$
iii.	Flatten $\boldsymbol{a_{ij}}$ into vector $\boldsymbol{x_i \in \mathbb{R}^n, i = 1, 2, \dots M, where\ M = K \times B, and\ D = w \times h \times d}$
iv.	Large *dilation _rate:* Coarse Features
v.	Small *dilation _rate:* Finer Features
vi.	Compare different configurations of dilation rates to find best-diagnosed results
vii.	Compute ReLU activation function, $\boldsymbol{f(x) = \begin{cases} x & for\ x \geq 0 \\ 0 & for\ x < 0 \end{cases}}$
viii.	Compute cross-channel normalization, $\boldsymbol{x' = \frac{x}{(K + \frac{\alpha \times ss}{window\ channel\ size})^{\beta}}}$ where $\boldsymbol{\alpha, \beta, K}$ are the hyperparameters in the normalization and ss = sum of squares of the elements in the normalization window
ix.	Apply max-out plan, $\boldsymbol{Z_s}$ to different feature maps $\boldsymbol{O_s, O_{s+1}, \dots \dots, O_{s+k-1}}$ which takes maximum over the $\boldsymbol{O_s}$ and maps it individually as represented in $\boldsymbol{Z_{s,i,j} = \max(O_{s,i,j}, O_{s+1,i,j}, \dots \dots, O_{s+k-1,i,j})}$
x.	Apply $\boldsymbol{Adam\_Optimizer}$ to minimize error rate
xi.	Repeat steps ii, vii, ix twice for both parallel paths where filter number 96.
xii.	Update weights using $\boldsymbol{back\_propagation}$
xiii.	$\boldsymbol{Weights_{Best}} \longleftarrow$ Save Weights
xiv.	Employ the optimized weights to extract the multiscale features in the training set

3.5.1 Multiscale Feature Selection Path

CNNs have been used extensively in the field of medicine and have demonstrated good results in the segmentation and classification of medical images. CNN architectures are built using a variety of building blocks, such as Fully-Connected (FC) layers, pooling layers, and convolution layers. Convolution layers, which combine linear and nonlinear operations—that is, activation functions and convolution operations—are used in feature extraction. Kernels and their hyper-parameters, such as the size, quantity, stride, padding, and activation function of each kernel, are the parameters of convolution layer. Six convolution layers are used in the two simultaneous paths and the convolution operation occurs using equation (3.1) [37].

$$O_{p,q,r}^{l} = f\left(W_r^l\ I_{p,q}^{l-1}\ +\ b_r^l\right) \tag{3.1}$$

Where, for r^{th} kernel in layer l, $O_{p,q,r}^{l}$ expresses the resultant feature map of position (p, q), W_r^l represents the weight vector's values, $I_{p,q}^{l-1}$ indicates the input vector of position (p, q) in the $l-1$, and b_r^l is the symbol of bias. In addition, the activation function is $f(.)$. By down-sampling, pooling layers lower the dimensionality of the feature maps. The stride, padding, and filter size are among the hyper-parameters that comprise pooling layers, although they do not contain any other parameters.

Two common varieties of pooling layers are max pooling and global average pooling. Maximum pooling layer is used in this structure. The output size of the pooling operation in CNN is calculated using equation (3.2).

$$O = \left[\frac{n-f+2p}{s}\right] + 1 \tag{3.2}$$

Where, n stands for the dimension of input, f is the kernel size, the padding size is shown by p, and s is symbol of stride size [40].

The pooling layers' feature maps are smoothed out and sent to several one-dimensional (1D) vectors known as FC layers. The most popular activation parameter for FC layers is the Rectified Linear Unit (ReLU), which is illustrated in (3.3) [38].

$$f(\boldsymbol{x}) = \begin{cases} x & for\ x \geq 0 \\ 0 & for\ x < 0 \end{cases} \tag{3.3}$$

The final FC layer's activation function is usually SoftMax for the categorization of multiple classes and Sigmoid for binary classification. The node values in the final FC layer of the proposed model has computed using (3.4), and the sigmoid activation function for a binary categorization dataset-I is calculated using (3.5) [37, 39].

$$z = w^T h + b \tag{3.4}$$

$$P(y = 1|x) = \frac{1}{1+exp(-z)} \tag{3.5}$$

Where, h stands for the neural network layers' internal calculations, b shows the bias, and w stands for the weights used to determine an output node's value. Furthermore, the input vector and output class are denoted by x and y, respectively. The SoftMax activation function is calculated using (3.6) for the

multi-class categorization Figshare dataset-II and Kaggle dataset-III in this proposed structure [40].

$$P(y|x) = \frac{exp(f_y)}{\sum_{c=1}^{C} exp(f_c)} \quad (3.6)$$

Where, x stands for the input vector and y for the class in the case of a multi-class categorization problem. Additionally, the c^{th} component of the class rating vector in the final FC layer is displayed by f_c. The category k with the highest P coefficient is chosen as the output class in the SoftMax activation function.

A backpropagation algorithm has used during CNN training to adjust the weights of the FC and convolution layers.

Expanding the receptive field in deep learning involves boosting the dimension and depth of the convolution kernel, which in turn enhances the number of elements in the network. By adding weights of zero to the conventional convolution kernel, dilated convolution may enhance the receptive field without adding more network elements.

Equation (3.7) defines the convolution operator * as follows: 1-D dilated convolution using dilation rate $l = 1$ connects input picture F alongside kernel k. The term "standard convolutional neural network" refers to the 1-D convolution. The network is identified as dilated convolutional neural network when the dilation rate l rises.

$$(F * k)(p) = \sum_{s+t=p} F(s) \times k(t) \quad (3.7)$$

Upon the introduction of a dilation rate when a dilation factor called l is introduced and by generalizing this factor l that can be defined as,

$$(F *_l k)(p) = \sum_{s+lt=p} F(s) \times k(t) \quad (3.8)$$

Using equation (3.8), the dilated convolution operation is calculated in this proposed structure. The fundamental CNN has a value of $l = 1$[42, 43].

The main function of dilated convolution layer is to extract features. In addition to conveying fine and high-level feature details, MRI images also contain rough and low-level information. As a result, image data must be extracted at several scales. Specifically, the local and global routes are employed

to obtain the local and global features. Within the local route, the convolutional layers make use of the small 5x5 pixel window dimension to provide low-level details about the images. However, a vast number of filters with 12x12 pixels are present in the convolutional stages of the global path. The same 5 by 5 filters are used by three different convolution layers throughout the local path, and each layer's decremented even number of high DF (4, 2, 1) is the only factor used to produce the coarse feature maps. Three distinct convolution layers in the global path employ identical 12 × 12 filters, and the generation of finer feature maps is exclusively dependent on the tiny DF (2, 1, 1) of every single layer. As illustrated in Fig. 3.3, three convolution layers with distinct filter numbers (128, 96, 96) are applied at each feature extraction path to extract image data at various scales.

Conv1, Conv3, and Conv4 provide local as well as coarse features, while Conv2, Conv5, and Conv6 supply global as well as fine features. The max-pooling layer is employed after each convolutional layer for each path that down-samples the output of the convolutional layer. By employing a 2 × 2 kernel, the max-pooling layers lower the dimension of the attributes that are produced.

A dimension of (32, 32, 1) is assigned to each input tensor in the suggested model's structure. To test the impact of the DF on the model's efficiency and comprehend the gridding impact brought about by the dilation approach, the interior design is kept as simple as possible. In the local path, layer Conv1 applies a 5 × 5 filter and a dilation factor of d_1=4 to generate coarse feature maps (such as shapes and contours); layer Conv3 applies the same filter and dilation factor of d_2=2 along with the final convolution to generate coarse feature maps once more; and layer Conv4 applies a 5 × 5 filter and dilation factor of d_3=1 to generate coarse feature maps. In the global route, layer Conv2 applies a 12 × 12 filter and a dilation factor of d_4=2, layer Conv5 applies the same filter and dilation factor of d_5=1 along with the last convolution to generate fine feature maps once more, and layer Conv6 applies a 12 × 12 filter and a dilation factor of

d_6=1 to generate fine feature maps. The activation function of ReLU is utilized by all six convolutional stages.

3.5.2 Merge Stage

A merge layer connects the two routes, creating a single path with a cascaded link until it reaches the endpoint. This process extracts multiscale features, where local paths with high dilation rates extract local as well as coarse features, and global paths with low dilation rates extract global as well as fine features. Two fully interconnected layers that are connected to a dropout layer through the merging pathway come after a batch normalization layer and a ReLU layer. This model has included dropout layer to handle the overfitting issues and speed up the training process. Overfitting occurs when a model gets highly good at capturing the intricate details of the training data, limiting its capacity to generalize efficiently to new, previously unknown data. This problem becomes more evident as the network's depth increases as layers increase, resulting in outstanding performance on the training set but a significant loss on the testing set. The proposed dilated PDCNN presented in this study is highlighted for its minimal number of layers and parameters. In order to address the issue of performance degradation brought on by a boost in neural network stages, batch normalization is used. Equation (3.9) provides the feature map that results after a merging phase [35].

$$Z = \sigma(BN(f(X))) \tag{3.9}$$

Where, σ stands for the ReLU activation function, BN denotes the batch normalization function and $f(X)$ denotes the fused feature maps from each channel in the preceding paths.

3.5.3 Hyper-parameter Tuning

Hyper-parameter adjusting is a successful parameter searching technique for the suggested dilated PDCNN framework. The dense layer, optimization, and dropout measure are among the parameters that must be chosen to perform this PDCNN adjustment.

It provides the framework with the ideal set of parameters, producing the most effective results. The training data for the simulated scenario is provided by the effective adjustment of the hyper-parameter, which includes the Adaptive Moment Estimation (Adam) optimizer, 0.3 dropout, 512 dense layers, and 0.0001 rate of learning. In this work, the weight of the layers is updated via Adam, the optimizer that calculates the adaptive learning rates of every parameter. The training setting employs a validation frequency of 20 Hz. The highest average accuracy for the test datasets is collected for each run. When the epoch count reaches 70, the framework is trained employing a range of epoch counts; it acquires 98.67% accuracy for dataset-I. It acquires 98.13% and 98.35% accuracy for dataset-II, and dataset-III respectively when the epoch number is 60.

Table 3.3. Hyper-parameter Settings for Model Training.

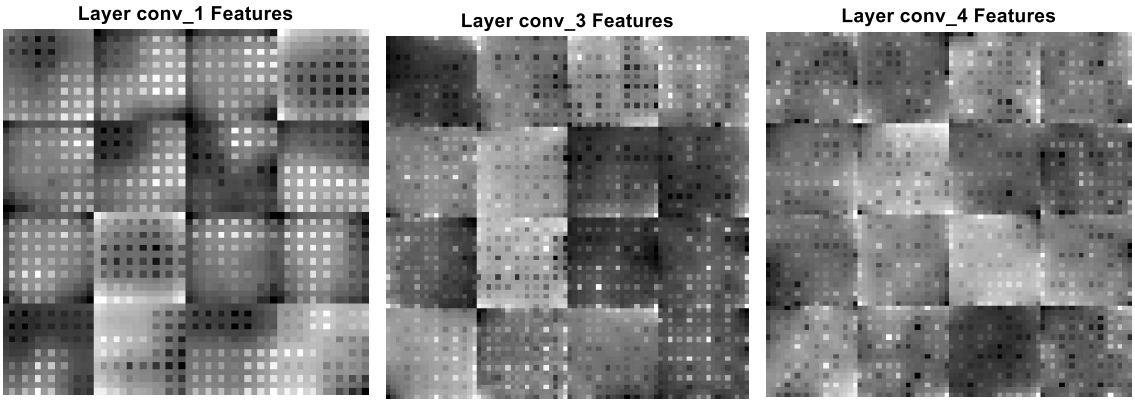
Hyper-parameter	Optimized Value
Optimizer	Adam
Dropout	0.3
Dense Layer	512
Learning Rate	0.0001
Maximum Epoch	50
Validation Frequency	20
Iteration Per Epoch	34

3.5.4 Feature Map of Dilated Convolutional Layer

CNN feature map represents specific attributes in the input image as the result of a convolutional layer. It is produced by filtering input images or the previous layers' feature map output. The feature maps that are acquired from every convolutional layer are presented in Fig. 3.4 and 3.5. In Fig. 3.4, the low-level and coarse features of the three convolutional layers conv_1, conv_3, and conv_4 having filters of 128, 96, and 96 are displayed. The feature maps in this figure are primarily composed of coarse and local features which represent the texture in an image. In this local path, a dilated CNN algorithm that has DFs associated with (d_1= 4, d_2= 2, d_3= 1) is referred to as dilated PDCNN (4, 2, 1). In Fig. 3.5, the high-level feature maps include contour representations, shape

descriptors and fine features of the deeper three convolutional layers conv_2, conv_5, and conv_6 having the same filters, are shown. DFs corresponding to (d_4=2, d_5= 1, d_6= 1) are used in this global path. The multiscale feature maps, which are displayed in Fig. 3.6, are greatly improved when these features are combined using a feature fusion technique. Fig. 3.7 displays the final multiscale features that are extracted, along with a fully connected layer that is prevented from overfitting by employing the dropout technique.

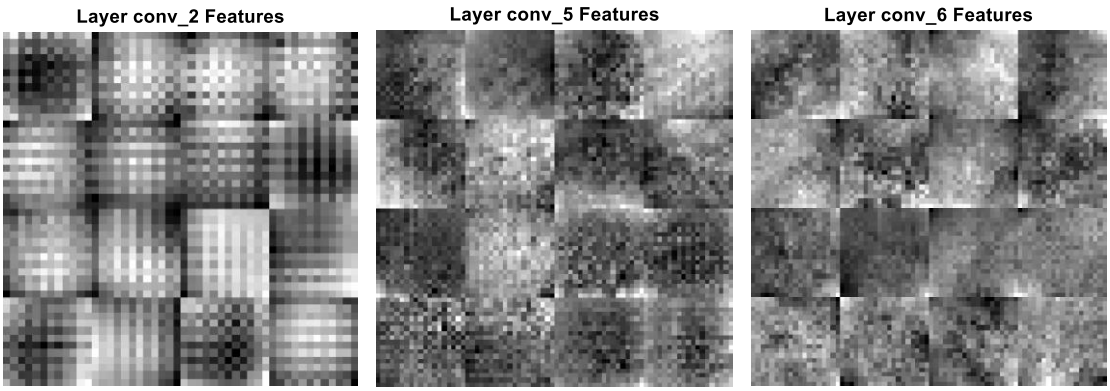
Local & Coarse Features (conv_1 [DF: 4*4], conv_3 [DF: 2*2], and conv_4 [DF: 1*1])



(a) (b) (c)

Fig. 3.4 Local and coarse feature maps of various convolutional layers of the Dilated PDCNN (a) Feature map of conv_1-layer (b) Feature map of conv_3-layer (c) Feature map of conv_4-layer.

Global & Finer Features (conv_2 [DF: 2*2], conv_5 [DF: 1*1], and conv_6 [DF: 1*1])



(a) (b) (c)

Fig. 3.5 Global and finer feature maps of various convolutional layers of the Dilated PDCNN (a) Feature map of conv_2-layer (b) Feature map of conv_5-layer (c) Feature map of conv_6-layer.

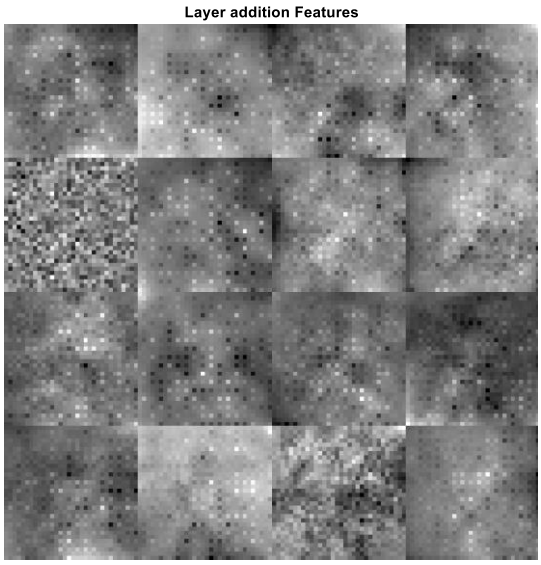


Fig. 3.6 Addition of all features.

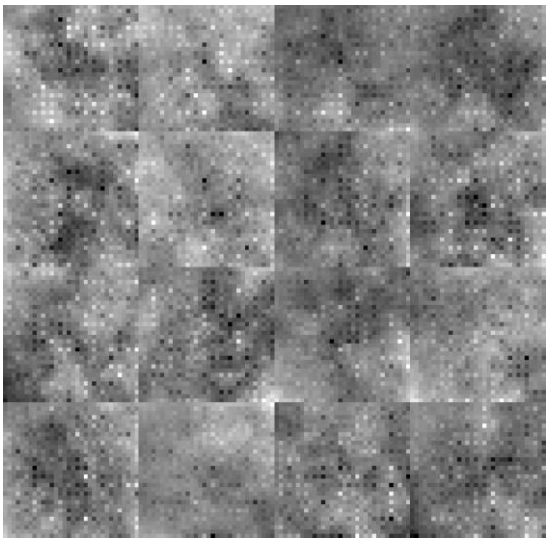


Fig. 3.7 Features extraction after FC_2 layer.

3.5.5 Parameters for Dilated PDCNN Model

Both FC and convolutional layers provide parameters that can be learned. Parameters are the quantities of weights that the CNN structure learns during training. It is possible to calculate the convolutional layer parameters (P_{conv}) equation as follows [10]:

$$P_{conv} = F_h * F_w * F_{num} * C_{in} + F_{num} \quad (3.10)$$

F_h and F_w stand for the length and width of the filter, accordingly. F_{num} indicates the quantity of filters. C_{in} indicates the associated layer's input channel quantity. The parameters of the layer that is fully connected (P_{FC}) are as follows [10]:

$$P_{FC} = A_{(prev)} * N_{(unit)} + N_{unit} \quad (3.11)$$

Where, $A_{(prev)}$ indicates the prior layer's activation pattern and $N_{(unit)}$ denotes the number of neurons that make up the present FC layer. There are no variables that can be learned in the max-pooling layer. The batch-normalization layer's variables are the product of the number of channels utilized in the preceding convolutional layer.

Table 3.4. Information about the Suggested Dilated PDCNN Framework.

Layer Type	**No of Filters**	**Filter Size**	**Stride**	**Dilation Factor**	**Activation Shape**	**Total Learnable Parameters**
Image MRI	-	-	-	-	32×32×1	0
Conv layer	128	5×5	2,2	4,4	16×16×128	3328
ReLU	-	-	-	-	16×16×128	0
Cross Channel Normalization	-	-	-	-	16×16×128	0
Max Pooling	-	2×2	2,2	-	8×8×128	0
Conv layer	96	5×5	2,2	2,2	4×4×96	307296
Conv layer	128	12×12	2,2	2,2	16×16×128	18560
ReLU	-	-	-	-	4×4×96	0
Max Pooling	-	2×2	2,2	-	2×2×96	0
Conv layer	96	5×5	2,2	1,1	1×1×96	230496
ReLU	-	-	-	-	1×1×96	0
Max Pooling	-	2×2	2,2	-	1×1×96	0
ReLU	-	-	-	-	16×16×128	0
Cross-Channel Normalization	-	-	-	-	16×16×128	0
Max Pooling	-	2×2	2,2	-	8×8×128	0
Conv layer	96	12×12	2,2	1,1	4×4×96	1769568
ReLU	-	-	-	-	4×4×96	0
Max	-	2×2	2,2	-	2×2×96	0
Conv layer	96	12×12	2,2	1,1	1×1×96	1327200
ReLU	-	-	-	-	1×1×96	0
Max Pooling	-	2×2	2,2	-	1×1×96	0
Elements-wise addition of 2 inputs	-	-	-	-	1×1×96	0
Batch Normalization	-	-	-	-	1×1×96	192

ReLU	-	-	-	-	1×1×96	0
FC_1 layer	-	-	-	-	1×1×512	49664
ReLU	-	-	-	-	1×1×512	0
Dropout layer	-	-	-	-	1×1×512	0
FC_2 layer	-	-	-	-	1×1×2	1026
						Total = 3, 707, 330

3.6 CLASSIFICATION STAGE

In this categorization phase, extracting all the multiscale attributes from the last FC layer, four types of classifiers: SVM, KNN, NB, and Decision tree are used to categorize the three types of brain tumor datasets. The following subsections go over the machine learning classifiers and their hyper-parameter settings that were employed in this experiment to classify brain tumors.

3.6.1 SVM

One of the most potent classification algorithms, SVM was introduced by Vapnik. It operates by building a hyperplane with the greatest possible margin between classes. SVM converts the initial data space into a higher dimension space using the kernel function, $K(x_n, x_i)$. The following (3.12) defines the hyperplane function that will be used to separate the data [45]:

$$f(x_i) = \sum_{n=1}^{N} \alpha_n y_n K(x_n, x_i) + b \tag{3.12}$$

Where, x_n is support vector data (deep features from brain MR image), α_n is Lagrange multiplier, and y_n represent a target class of these three datasets employed in this research, such that the first dataset is binary (normal and abnormal) class dataset, while the second dataset has three classes (glioma, meningioma, and pituitary) and the third dataset has four different classes (normal, glioma, meningioma, and pituitary tumor) with n = 1, 2, 3, ..., N. In this work, the most commonly used kernel function is used at the SVM algorithm is linear kernel.

As shown in (3.13), an actually separable case is handled using a linear kernel.

$$\text{Linear: } K(x_n, x_i) = (x_n, x_i) \tag{3.13}$$

Where, $K(x_n, x_i)$ represents kernel function [45, 48].

Hence, a multiclass linear support vector machine with a linear kernel function and a zero verbose value is employed in this suggested method.

It is apparent that the SVM classifier is added at the final stage of the suggested model's output. The effective features that are produced by applying the suggested model serve as the inputs for the SVM classifier. Three different types of output are present in the SVM classifier because this method uses three different types of datasets. Table 3.5 lists the SVM network's parameters.

Table 3.5. SVM Classifier Parameter.

SVM Parameters	Size/ Value
C	1.0
Filter/kernel	"linear"
Degree	3
probability	"false"
Shrin king	true
class weight	"none"
verbose	false
Random state	"none"
Decision function shape	"ovr"

3.6.2 K-NN

One of the most basic approaches for classification is K-NN. It performs predictions directly from the training set that is stored in the memory. For instance, to classify a new data instance (a deep feature from brain MR image), k-NN chooses the set of k objects from the training instances that are closest to the new data instance by calculating the distance and assigns the label with two classes (normal or tumor), three classes (glioma, meningioma, and pituitary) or four classes (normal, glioma, meningioma, and pituitary tumor) and does the selection based on the majority vote of its k neighbors to the new data instance.

The distances between the new data instance and the training data instances are the most frequently measured using Manhattan distance and Euclidean distance. The k-NN algorithm in this work, is measured using the Euclidean distance. Data points x and y's Euclidean distance, d are computed as follows [48]:

$$d(x, y) = \sqrt{\sum_{i=1}^{N}(x_i - y_i)^2} \tag{3.14}$$

Below is an illustration of the k-NN algorithm in brief [48]:

- Chose a suitable distance metric initially.
- During the training stage, store all the training data set P in pairs as follows:

  $$P = (y_i, c_i), i = 1, \ldots, n \tag{3.15}$$

  where in the training dataset, y_i is a training pattern, n is the amount of training patterns and c_i is its corresponding class.
- Calculate the distances between the new features vector and the stored (training data) features during the testing stage. Then, classify the new class example by based on the vote of the k neighbors who have the most votes.

The accuracy of the algorithm is assessed using the correct classification provided during the test phase. The k value can be changed until a respectable degree of accuracy is achieved if the outcome is not satisfactory. As can be seen, the number of neighbors is limited to 5, and selected the best value of K was determined to be 5. This value is then applied with the highest accuracy to dataset-I, II, and III, respectively. The zero standardized value is used to standardize the predictors.

It is apparent that the K-NN classifier is added at the final stage of the suggested model's output. The effective features that are produced by applying the suggested model serve as the inputs for this classifier. Table 3.6 lists the K-NN network's parameters.

Table 3.6. K-NN Classifier Parameter.

KNN Parameters	**Size/Value**
Leaf size	30
N_neighbors	5
Metric params	"none"
weights	"uniform"
p	2
algorithm	"auto"

3.6.3 Naïve Bayes

NB classifier is the machine learning classifier with the assumption of conditional independence between the attributes given the class. In this article, the final class is predicted using vectors of attributes and class prior probabilities, as shown in (3.16) [48].

$$P\ (C = i \mid X = x) = \frac{P\ (X=x\ |C=i)*P(C=i)}{P\ (X=x)} \tag{3.16}$$

Where, X indicates the given data instance (extracted deep features from brain MR image) which is represented by its feature vector $(x_1, \dots, x_n)$ and C is the class target (type of brain tumor) with two classes (normal and tumor) for binary dataset-I or three classes (glioma, meningioma, and pituitary) for Figshare dataset-II, or four classes (normal, glioma, meningioma, and pituitary) for multiclass Kaggle dataset-III . Here, $P\ (X = x \mid C = i)$ in this classifier is computed by taking the dataset's features to be independent, and calculating the probability as shown in (3.17) [48, 49].

$$P\ (X = x|C = i\ ) = \prod_{j=1}^{n} P\ \left(x_j\ \middle|C = i\right) \tag{3.17}$$

NB can save a significant amount of time and is appropriate for handling multi-class prediction problems. The value for the minimum threshold of probabilities for the NB classifier is 0.001. For every predictor, and class combination, the kernel smoothing window width is automatically chosen by default in this suggested approach.

3.6.4 Decision Tree

Decision tree structures represent an approach to non-parametric supervised learning that can be employed in classification as well as regression. This model is employed in this research to create a model that makes use of fundamental decision rules inferred from the data features taken out of brain magnetic resonance imaging. With training vectors $x_i \in \mathbb{R}^n$, i= 1,…,l and a label vector $y \in \mathbb{R}^l$, a decision tree successively divides the domain of features so that instances that share similar desired values are organized together.

Three datasets are used in this proposed approach. Here dataset-I at node m be denoted by Q_m. Splitting $\theta = (j, t_m)$ for every candidate which is composed of a characteristic j and threshold t_m, dividing the dataset-I,Q_m into $Q_m^{left}(\theta)$ and $Q_m^{right}(\theta)$ subgroups [48]

$$Q_m^{left}(\theta) = \{(x, y) | \, x_j \leq t_m\} \tag{3.18}$$

$$Q_m^{right}(\theta) = Q_m \setminus Q_m^{left}(\theta) \tag{3.19}$$

Next, to calculate the quality of a potential split of node m, a loss function H() is used [48].

$$G(Q_m, \theta) = \frac{n_m^{left}}{n_m} H\left(Q_m^{left}(\theta)\right) + \frac{n_m^{right}}{n_m} H(Q_m^{right}(\theta)) \tag{3.20}$$

This process should be carried out for $Q_m^{left}(\theta^*)$ and $Q_m^{right}(\theta^*)$ subsets until the maximum permitted depth is achieved. These steps are also followed for the other two datasets multiclass Figshare dataset-II and Kaggle dataset-III.

The decision tree can quickly determine which data is relevant and which is not. The maximum number of branch nodes in the suggested method is fixed at 1. In this approach, there must be a minimum of one leaf node.

3.6.5 Average Ensemble Method

Machine learning and signal analysis both use the statistical technique of ensemble averaging. Model averaging is a machine learning technique for ensemble learning in which each member of the ensemble makes an equal contribution to the ultimate prediction. A group of frameworks often

outperforms a single one because the individual errors in the models "average out." In average ensemble method, the actions are:

- Create N experts, each starting at a different value: Typically, initial values are selected at random from a distribution.
- Train every specialist independently.
- Add up all the experts and take the mean of their scores.

These steps of average ensemble method are used to average the accuracy to acquire the final outcome in the proposed method [51, 52].

A comprehensive flowchart for the methodological framework is illustrated in this chapter for providing a concise understanding on how the research is performed. Details on designing and modeling of various algorithms along with dataset description, dataset preprocessing, data augmentation, and design parameters and figures are discussed in the chapter.

Chapter 4: RESULT AND ANALYSIS

This chapter investigates how the suggested system performs with various dilation factors, classifiers, and the feature fusion technology. Section 4.1 describes the experimental outcomes and evaluation of the proposed system including the performance analysis of the suggested dilated PDCHH model, comparative analysis of different dilation rate, impact of applying dilation on the proposed model, evaluation measurements of the proposed system on the three datasets and analysis between the proposed model and earlier research on the topic.

4.1 EXPERIMENTAL OUTCOMES AND EVALUATION

MATLAB is used to run the implementation program for the suggested dilated PDCNN model. The computing device is equipped with a Core i5 processor manufactured by Intel, running at 3.2 GHz, eight GB of RAM, and Windows 10 operating system installed.

4.1.1 Performance Analysis of Suggested Dilated PDCNN Model

The confusion matrix is used to express the classification system's results. The efficiency is evaluated using the following criterion [52].

$$Accuracy = \frac{True\ Positive+True\ Negative}{(Positive+Negative)} \tag{4.1}$$

$$Precision = \frac{True\ Positive}{(True\ Positive+False\ Positive)} \tag{4.2}$$

$$Recall = \frac{True\ Negative}{(False\ Positive+True\ Negative)} \tag{4.3}$$

$$F1\ Score = \frac{2*True\ Positive}{(2*True\ Positive+False\ Positive+False\ Negative)} \tag{4.4}$$

Table 4.1 provides an overview of the effectiveness statistics of the PDCNN strategy using ML classifiers for dataset-I. Comparing the PDCNN algorithm using the Decision Tree classifier to the remaining models, Table 4.1 shows that it achieves the greatest level of accuracy, precision, recall, and F1 score. With a 98.67% accuracy rate, 96.55% precision rate, 97.87% recall rate, and 98.25% F1-

score, the PDCNN model utilizing the Decision Tree classifier outperforms comparable machine learning classifiers. In conclusion, the PDCNN's accuracy, precision, recall, and f1-score when using the average ensemble technique are 96.54%, 94.16%, 96.37%, and 95.49%, respectively.

Table 4.1. Performance Parameters of PDCNN Model with ML Classifiers on Dataset-I.

PDCNN Models Utilizing ML Classification	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**
PDCNN	96.03	94.92	96.74	94.92
PDCNN with SVM	97.33	93.10	95.83	96.43
PDCNN with KNN	96.00	93.10	95.74	94.74
PDCNN with NB	94.67	93.10	95.65	93.10
PDCNN with Decision Tree	98.67	96.55	97.87	98.25
PDCNN with Average Ensemble	**96.54**	**94.16**	**96.37**	**95.49**

Table 4.2. Performance Parameters of Dilated PDCNN Model with ML Classifiers on Dataset-I.

Dilated PDCNN Models Utilizing ML Classification	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**
Dilated PDCNN	97.33	93.10	95.83	96.43
Dilated PDCNN with SVM	98.67	100.00	100.00	98.31
Dilated PDCNN with KNN	100.00	100.00	100.00	100.00
Dilated PDCNN with NB	97.33	100.00	100.00	96.67
Dilated PDCNN with Decision Tree	100.00	100.00	100.00	100.00
Dilated PDCNN with Average Ensemble	**98.67**	**98.62**	**99.17**	**98.28**

Table 4.2 provides an overview of the dilated PDCNN algorithm's efficiency indicators using ML classifiers over dataset-I. As per Table 4.2 findings, the dilated PDCNN model that incorporates KNN and Decision Tree classifiers has the best F1-score, recall, accuracy, and precision in comparison with the remaining models. The suggested dilated PDCNN model utilizing KNN and Decision Tree classifiers has 100.00% for all performance criteria, which is better than the outcomes obtained by other ML classifiers. The dilated PDCNN model that includes SVM and NB classifiers is noteworthy for having 100% precision and recall, which is also the same as the dilated PDCNN model alongside KNN

and Decision Tree classifiers. However, the KNN and Decision Tree classifier execute better concerning other performance indicators. The suggested dilated PDCNN utilizing the average ensemble approach for the dataset-I has finalized accuracy, precision, recall, and f1-score values of 98.67%, 98.62%, 99.17%, and 98.28%, accordingly, after implementing the average ensemble technique.

Table 4.3. Performance Parameters of PDCNN Model with ML Classifiers on Dataset-II.

PDCNN Models Utilizing ML Classification	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**
PDCNN	97.60	97.00	97.60	97.30
PDCNN with SVM	97.71	97.33	97.33	97.33
PDCNN with KNN	97.40	96.67	97.33	97.00
PDCNN with NB	97.40	97.00	97.00	97.33
PDCNN with Decision Tree	96.60	95.67	96.67	96.33
PDCNN with Average Ensemble	**97.34**	**96.74**	**97.19**	**97.06**

Table 4.3 provides an overview of the PDCNN algorithm's efficiency indicators using ML classifiers over dataset-II. The results shown in Table 4.3 show that, in comparison to other designs, the PDCNN model employing SVM provides the best accuracy, precision, and F1 score. In comparison to the outcomes of other ML classification methods, the PDCNN algorithm incorporating the SVM classifier offers an accuracy of 97.71%, precision of 97.33%, and F1-score of 97.33%. But the PDCNN has the highest recall value among all the models which is 97.60%. Finally, applying the average ensemble method, the values of accuracy, precision, recall, and f1-score of the PDCNN with average ensemble method for dataset-II are 97.34%, 96.74%, 97.19%, and 97.06% respectively which is greater than the method for dataset-I.

An overview of the dilated PDCNN algorithm's efficiency indicators using ML classifiers over dataset-II is provided in Table 4.4. The results shown in Table 4.4 demonstrate that, when contrasted to other scenarios, the dilated PDCNN algorithm using the NB classifier provides the highest performance indicators. The suggested dilated PDCNN model incorporating the NB classifier outperforms the findings of the remaining ML classifiers alongside accuracy of

98.90%, precision of 98.67%, recall of 98.67%, and F1-score of 98.67%. In the end, using the average ensemble approach, the suggested dilated PDCNN employing the average ensemble technique for the dataset-II has accuracy, precision, recall, and f1-score values of 98.13%, 97.74%, 98.05%, and 97.80%, accordingly.

Table 4.4. Performance Parameters of Dilated PDCNN Model with ML Classifiers on Dataset-II.

Dilated PDCNN Models Utilizing ML Classification	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**
Dilated PDCNN	98.20	98.00	98.33	98.00
Dilated PDCNN with SVM	97.72	97.33	97.33	97.33
Dilated PDCNN with KNN	97.60	97.00	97.60	97.30
Dilated PDCNN with NB	98.90	98.67	98.67	98.67
Dilated PDCNN with Decision Tree	98.21	97.67	98.33	97.67
Dilated PDCNN with Average Ensemble	**98.13**	**97.74**	**98.05**	**97.80**

Table 4.5. Performance Parameters of PDCNN Model with ML Classifiers on Dataset-III.

PDCNN Models Utilizing ML Classification	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**
PDCNN	96.80	96.25	96.50	96.25
PDCNN with SVM	97.94	98.00	97.50	97.75
PDCNN with KNN	97.80	97.50	97.50	97.25
PDCNN with NB	97.90	97.75	97.50	97.70
PDCNN with Decision Tree	97.40	97.75	96.50	96.75
PDCNN with Average Ensemble	**97.58**	**97.45**	**97.10**	**97.15**

Table 4.5 provides an overview of the PDCNN model's effectiveness indicators using the machine learning classifiers for dataset-III. The PDCNN algorithm utilizing SVM delivers the best-fit, performance indicators when compared to other scenarios, as indicated by the results shown in Table 4.5. With an accuracy rate of 97.94%, a precision of 98.00%, a recall of 97.50%, and an F1-score of 97.75%, the PDCNN model incorporating the SVM classifier outperforms other ML algorithms. Recall of the recommended PDCNN framework using KNN and NB classifiers is noteworthy at 97.50%, matching that of the suggested dilated PDCNN structure alongside the SVM classifier; however, the KNN

classifier performs better than all other classifiers in terms of other indicators of performance. Upon implementing the average ensemble method, the PDCNN's accuracy, precision, recall, and f1-score for dataset-III are 97.58%, 97.45%, 97.10%, and 97.15%, respectively.

Table 4.6. Performance Parameters of Dilated PDCNN Model with ML Classifiers on Dataset-III.

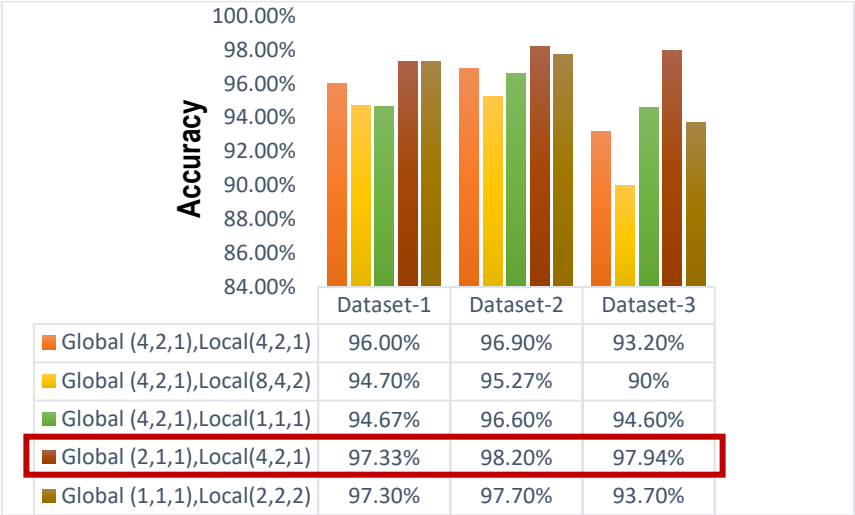
Dilated PDCNN Models Utilizing ML Classification	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Dilated PDCNN	98.21	98.25	97.75	98.25
Dilated PDCNN with SVM	98.60	98.50	98.25	98.50
Dilated PDCNN with KNN	98.50	98.50	98.00	98.50
Dilated PDCNN with NB	98.57	98.50	98.00	98.50
Dilated PDCNN with Decision Tree	97.85	98.00	97.25	97.25
Dilated PDCNN with Average Ensemble	**98.35**	**98.35**	**97.85**	**98.20**

Table 4.6 provides an overview of the effectiveness measurements of the suggested dilated PDCNN model using machine learning classifiers for dataset-III. In comparison to other models, the accuracy, precision, recall, and f1-score of the suggested dilated PDCNN alongside SVM classifier are 98.50%, 98.50%, 98.25%, and 98.50%, respectively, based on the results shown in Table 4.6. The findings of the dilated PDCNN employing average ensemble technique for dataset-⅖ are, after executing the average ensemble strategy, 98.35%, 98.35%, 97.85%, and 98.20%, accordingly, in terms of accuracy, precision, recall, and f1-score.

4.1.2 Comparative Analysis of Different Dilation Rate

While dilated convolution retains data resolution at the output layer and increases the receptive field without adding computation, stacking several dilated convolutions has the drawback of producing a grid effect. Validating the results involves comparing multiple combinations of dilation rates for the various convolution layers. Large dilation rates may impact tiny object recognition. As a result, the DF has gradually decreased (even-numbered arithmetic decreasing) at the local path in the suggested framework. By doing

this, the dilated feature map's sparsity is reduced, and more data can be extracted from the investigated region. At the global path, the low DF (2, 1, 1) has been carried out to extract the fine features.

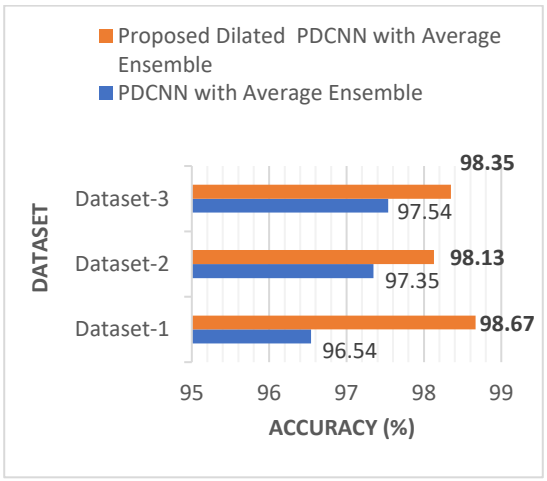


	Dataset-1	Dataset-2	Dataset-3
Global (4,2,1),Local(4,2,1)	96.00%	96.90%	93.20%
Global (4,2,1),Local(8,4,2)	94.70%	95.27%	90%
Global (4,2,1),Local(1,1,1)	94.67%	96.60%	94.60%
Global (2,1,1),Local(4,2,1)	97.33%	98.20%	97.94%
Global (1,1,1),Local(2,2,2)	97.30%	97.70%	93.70%

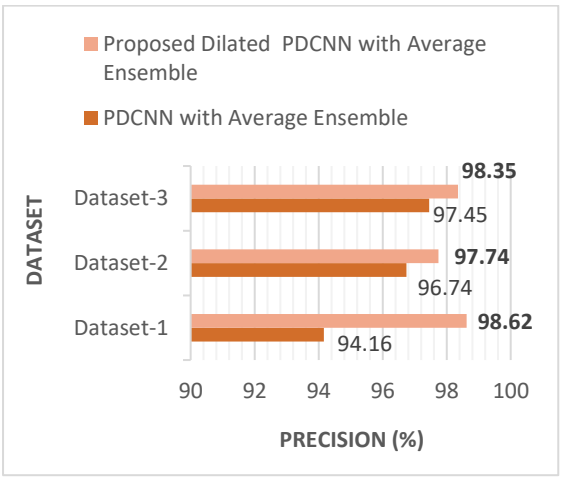
Fig. 4.1 Comparing accuracy across different configurations of dilation rates.

A comprehensive review of the gridding issue and the consequences of different dilation rates can be found in the accompanying Fig. 4.1. The poor efficiency of the (4, 2, 1) dilated value for the global pathway as well as (8, 4, 2) dilated value for the local route of the suggested model is caused by the gridding phenomenon, which arises when high DF are used. This limits the framework from acquiring finer characteristics. When a high DF (4, 2, 1) is used for both local and global paths, the accuracy increases more than before. On the contrary, using low dilation rates, the model only learns fine features. When low DF (1, 1, 1) is used in the global path and (2, 2, 2) is used for the local path, the value of accuracy for dataset-I, dataset-II, and dataset-III is 97.30%, 97.70%, and 93.70% respectively. When a low DF (2, 1, 1) is used for the global feature and a high DF (4, 2, 1) is used for the local feature, the highest accuracy is achieved. The highest accuracy for dataset-I, dataset-II, and dataset-III is 97.33%, 98.20%, and 97.94% respectively. Providing the best-case scenario, a well-balanced model (4, 2, 1) for the local path and (2, 1, 1) for the global path may acquire both the coarse as well as fine characteristics of the pictures.

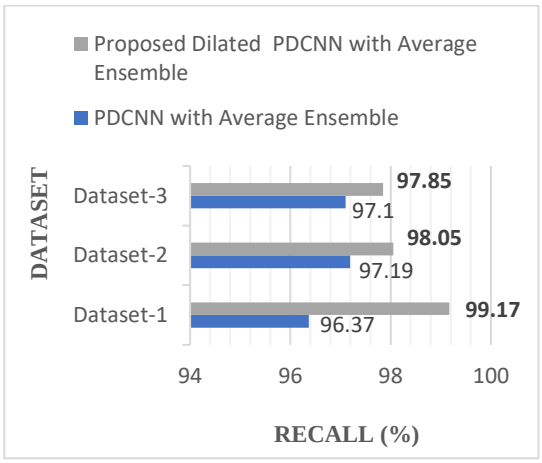
4.1.3 Impact of Applying Dilation on the Proposed Model



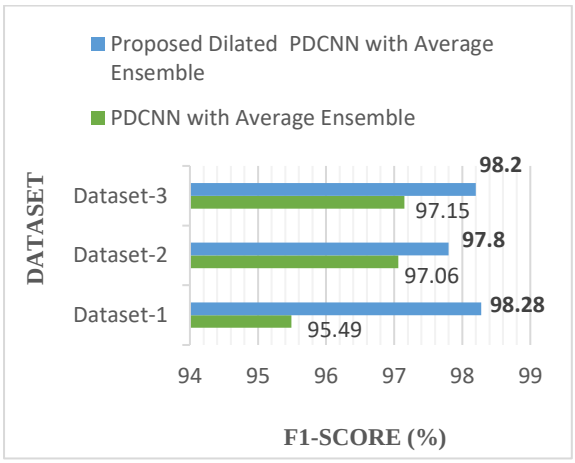
(a) Accuracy



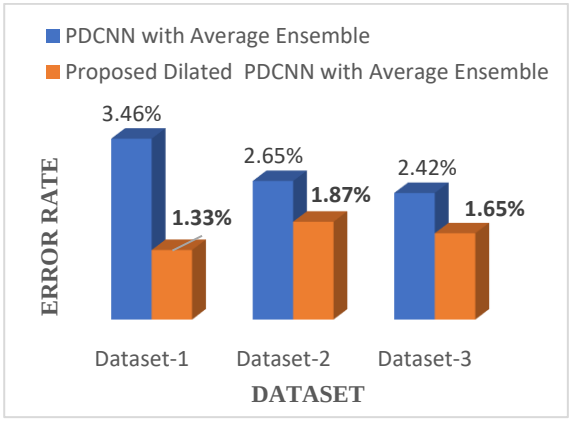
(b) Precision



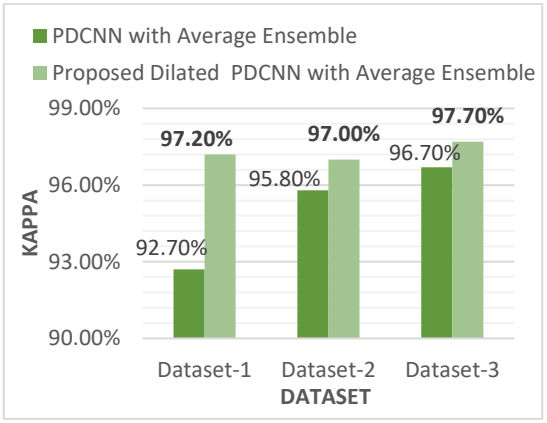
(c) Recall



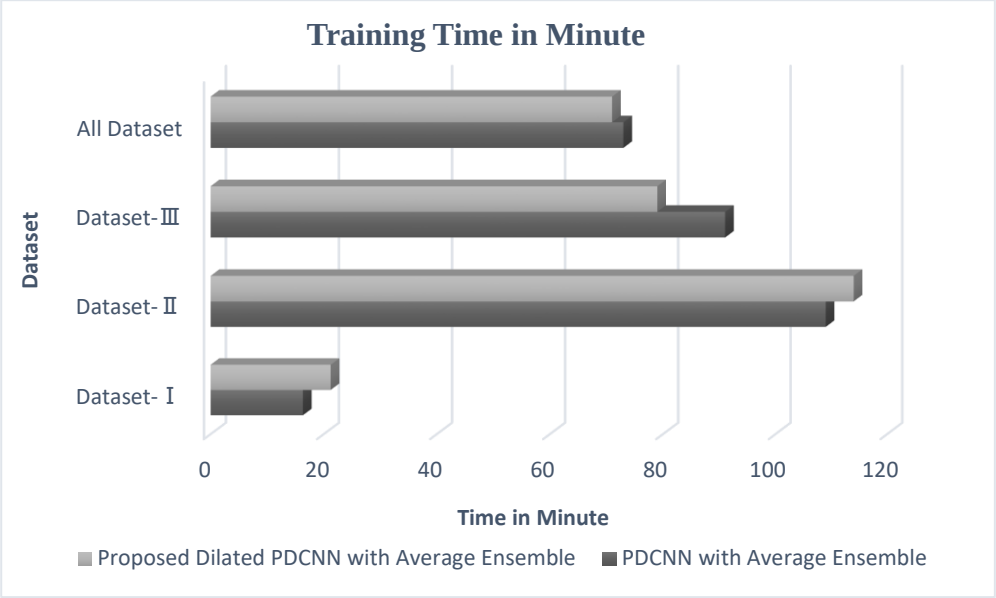
(d) F1-score



(e) Error Rate



(f) Kappa



(g) Training Time

Fig.4.2 Graphical representation of performance parameters of the proposed dilated PDCNN model.

In the categories of efficiency, precision, recall, F1 score, error rate, kappa, and training time, Fig. 4.2 shows that the suggested dilated PDCNN alongside the average ensemble approach executes better than the conventional PDCNN alongside the average ensemble framework. Values of the effectiveness indicators will increase even further if dilation is applied to increase the efficiency of the recommended approach. These findings show that in comparison to the proposed PDCNN model using the average ensemble technique, the proposed dilated average ensemble classifier for three types of dataset indicates a higher accuracy, precision, recall, f1-score, kappa, and lower error rate, training time.

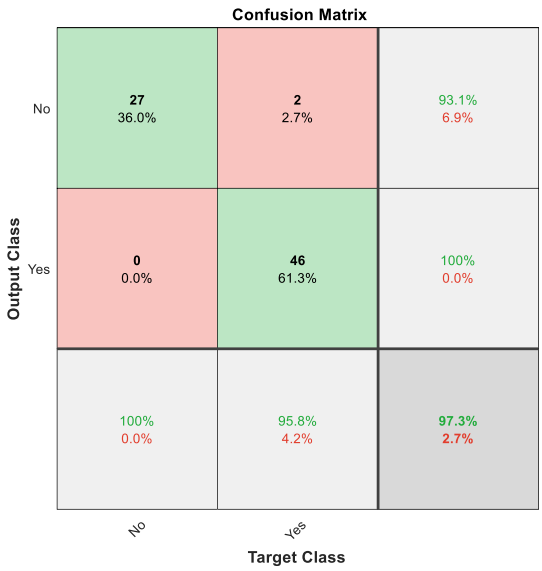
4.1.4 Evaluation Measurements of the Proposed System

With the SVM, KNN, NB, and Decision Tree classifiers for dataset-I, Table 4.7 illustrates the classification accuracy, erroneous, duration, and kappa scores for the suggested PDCNN as well as dilated PDCNN architectures. When the expected precision of the random classifier is considered, the kappa statistic expresses how closely the instances identified by the classification model matched the data assigned as ground truth. In comparison to the PDCNN alongside the average ensemble model, the dilated PDCNN has a larger kappa.

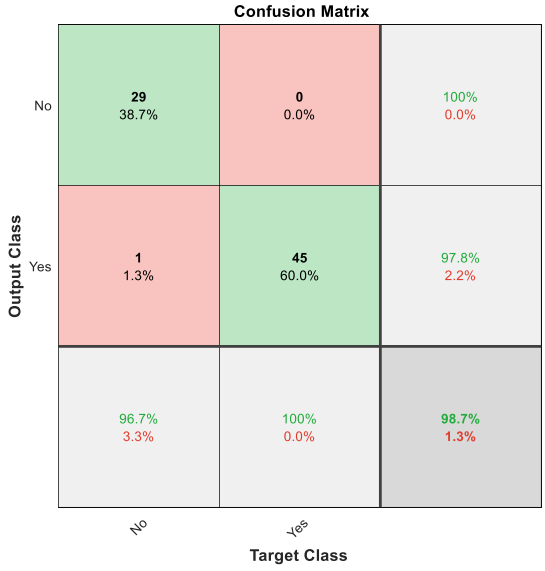
The error rate has reduced, and the elapsed time has increased following the application of dilation to the PDCNN with the average ensemble model.

Table 4.7. Evaluation Results of the Proposed System on the Binary Classification Dataset-I.

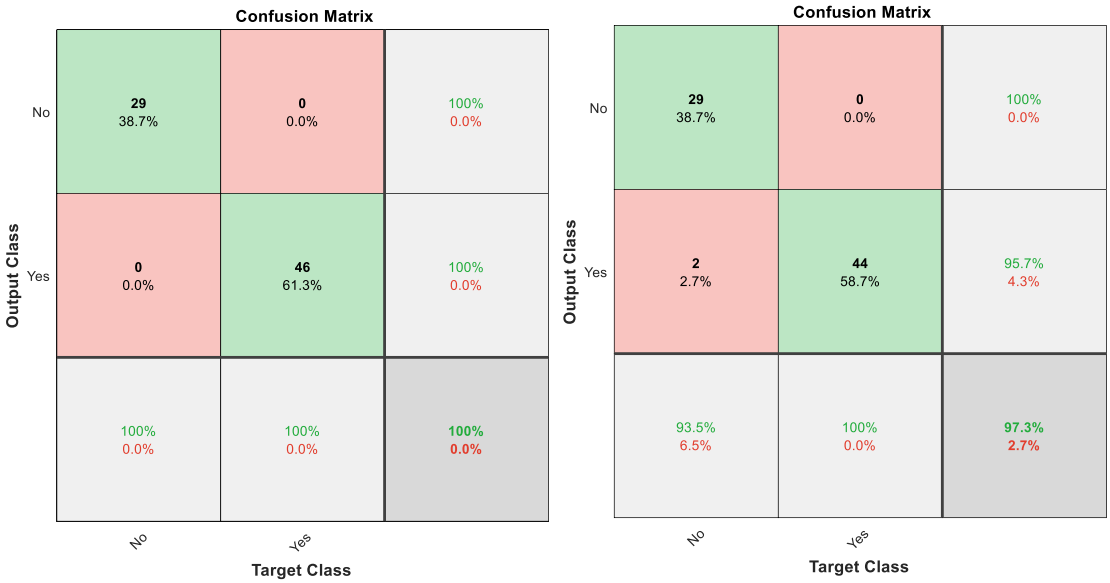
Structure	**Classifier**	Performance Indicators			
		Accuracy (%)	**Error (%)**	**Time (s)**	**Kappa**
PDCNN	Custom PDCNN	96.03	3.97	662	0.917
	PDCNN and SVM	97.33	2.67	1020	0.943
	PDCNN and KNN	96.00	4.00	1069	0.915
	PDCNN and NB	94.67	5.33	1079	0.888
	PDCNN and Decision Tree	98.67	1.33	1070	0.972
	Average Ensemble	**96.54**	**3.46**	**980**	**0.927**
Dilated PDCNN	Custom PDCNN	97.33	2.67	1683	0.943
	PDCNN and SVM	98.67	1.33	1020	0.972
	PDCNN and KNN	100.00	0.00	1223	1.000
	PDCNN and NB	97.33	2.67	1223	0.944
	PDCNN and Decision Tree	100.00	0.00	1223	1.000
	Average Ensemble	**98.67**	**1.33**	**1274**	**0.972**



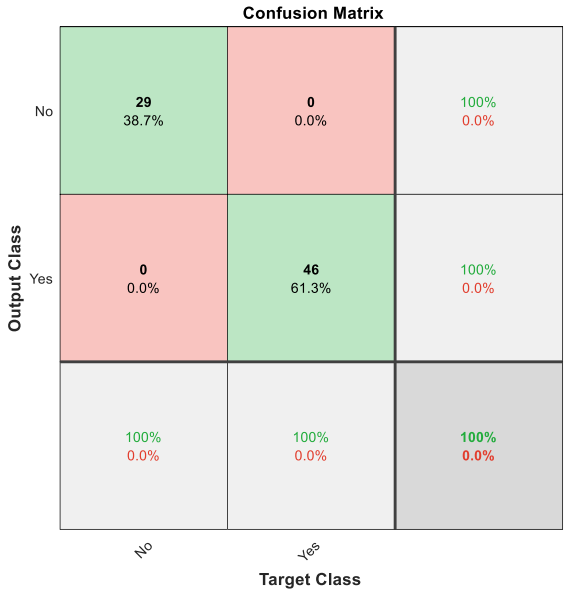
(a) Dilated PDCNN



(b) Dilated PDCNN + SVM



(c) Dilated PDCNN + KNN (d) Dilated PDCNN + NB



(e) Dilated PDCNN + Decision Tree

Fig.4.3 The suggested Dilated PDCNN model's confusion matrices employing ML classifiers for Dataset-I.

Fig. 4.3 displays the confusion matrices of the suggested dilated PDCNN model that includes ML classifiers. Comparing the recommended dilated PDCNN structure to the PDCNN model, this figure indicates that the accuracy of the latter is enhanced after dilation. When dilation is considered, the accuracy of the suggested model increases to 98.67% from 96.54% after using the ensemble technique.

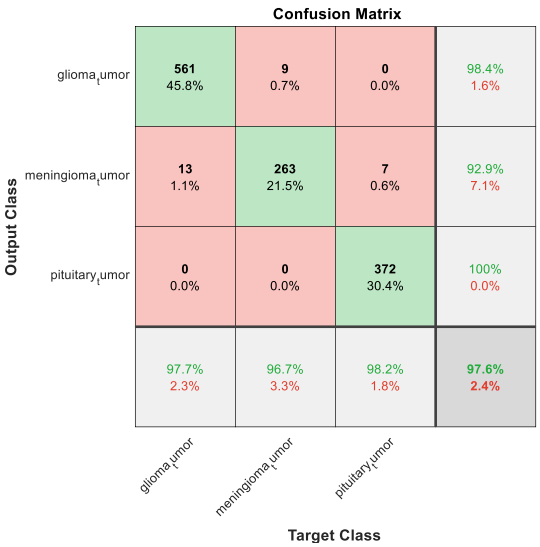
Table 4.8. Evaluation Results of the Proposed System on the Multiclass Figshare Dataset-II.

Structure	**Classifier**	Performance Indicators			
		Accuracy (%)	**Error (%)**	**Time (s)**	**Kappa**
PDCNN	Custom PDCNN	97.64	2.36	8050	0.963
	PDCNN and SVM	97.71	2.29	6106	0.960
	PDCNN and KNN	97.40	2.60	6307	0.959
	PDCNN and NB	97.40	2.60	4998	0.959
	PDCNN and Decision Tree	96.60	3.40	7170	0.946
	Average Ensemble	**97.35**	**2.65**	**6526**	**0.958**
Dilated PDCNN	Custom PDCNN	98.20	1.80	7204	0.972
	PDCNN and SVM	97.72	2.28	6106	0.962
	PDCNN and KNN	97.60	2.40	6187	0.961
	PDCNN and NB	98.90	1.10	6149	0.982
	PDCNN and Decision Tree	98.21	1.79	8381	0.972
	Average Ensemble	**98.13**	**1.87**	**6805**	**0.970**

The success rate, error, period, and kappa statistics for the suggested PDCNN and dilated PDCNN architectures employing the SVM, KNN, NB, and Decision Tree classifiers for dataset-II are presented in Table 4.8. When the expected precision of the random classifier is considered, the kappa statistic expresses how closely the instances identified by the classification model matched the data assigned as ground truth. As compared to the PDCNN employing the average ensemble model's kappa, the dilated PDCNN offers greater kappa. The error rate has dropped when dilation is applied to the PDCNN using the average ensemble method, but the time that passed has increased.

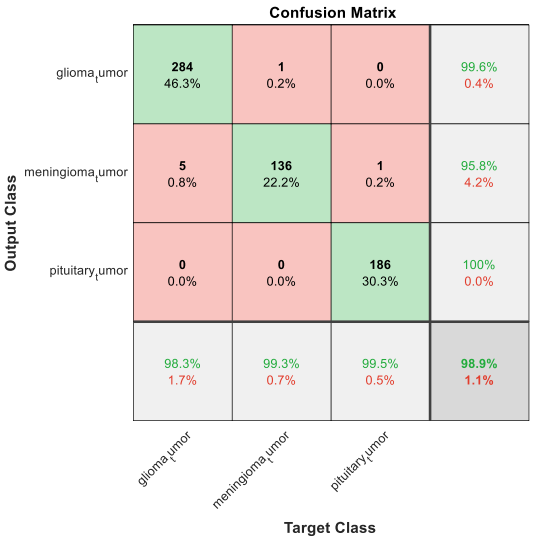
The suggested dilated PDCNN model containing ML classifiers' confusion matrices appears in Fig. 4.4. In contrast to the PDCNN model, this figure demonstrates that the accuracy of the recommended dilated PDCNN model increases after dilation. The accuracy of the suggested model increases to 98.13% from 97.35% after using the average ensemble method when dilation is considered.

(a) Dilated PDCNN



(b) Dilated PDCNN + SVM

(c) Dilated PDCNN + KNN



(d) Dilated PDCNN + NB

(e) Dilated PDCNN + Decision Tree

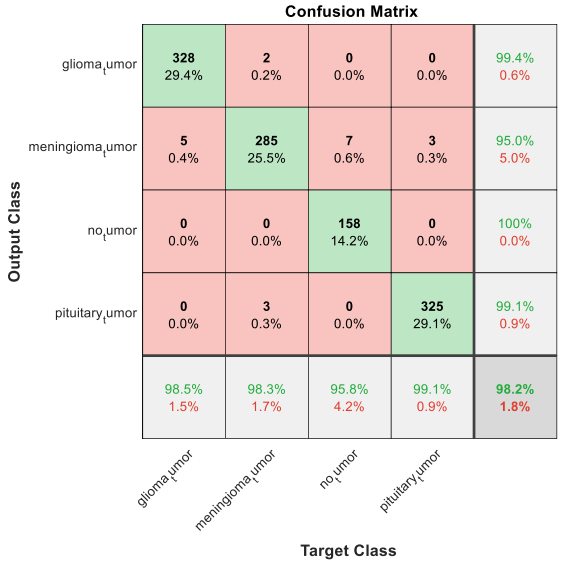
Fig.4.4 The suggested Dilated PDCNN model's confusion matrices employing ML classifiers for Dataset-II.

Table 4.9. Evaluation Results of the Proposed System on the Multiclass Kaggle Dataset-III.

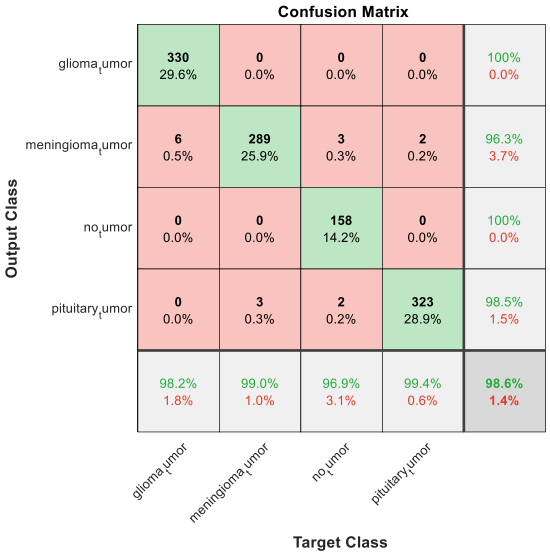
Structure	**Classifier**	Performance Indicators			
		Accuracy (%)	**Error (%)**	**Time (s)**	**Kappa**
PDCNN	Custom PDCNN	96.80	3.20	5633	0.956
	PDCNN and SVM	97.94	2.06	4462	0.972
	PDCNN and KNN	97.80	2.20	5753	0.969
	PDCNN and NB	97.90	2.10	5753	0.972
	PDCNN and Decision Tree	97.40	2.60	5753	0.965
	Average Ensemble	**97.58**	**2.42**	**5470**	**0.967**
Dilated PDCNN	Custom PDCNN	98.21	1.79	4891	0.976
	PDCNN and SVM	98.60	1.40	4739	0.980
	PDCNN and KNN	98.50	1.50	4739	0.979
	PDCNN and NB	98.57	1.43	4739	0.980
	PDCNN and Decision Tree	97.85	2.15	4739	0.971
	Average Ensemble	**98.35**	**1.65**	**4769**	**0.977**

Table 4.9 presents the kappa values, accuracy, error, and training duration for the recommended PDCNN and dilated PDCNN models that employ

the SVM, KNN, NB, and Decision Tree classifiers for dataset-III, in that order. When compared to the PDCNN employing an average ensemble model, the dilated PDCNN has a higher kappa value. The error rate has reduced, and the elapsed time has increased following the application of dilation to the PDCNN with the average ensemble model.



(a) Dilated PDCNN



(b) Dilated PDCNN + SVM

(c) Dilated PDCNN + KNN

(d) Dilated PDCNN + NB

(e) Dilated PDCNN + Decision Tree

Fig. 4.5 The suggested Dilated PDCNN model's confusion matrices employing ML classifiers for Dataset-III.

Fig. 4.5 displays the confusion matrices of the suggested dilated PDCNN model that includes ML classifiers. In comparison to the PDCNN approach, this figure demonstrates that the accuracy of the proposed dilated PDCNN structure increases after dilation. Implementing the average ensemble method while dilation is considered causes the accuracy of the suggested framework to increase from 97.58% to 98.35%.

4.1.5 Analysis between the Proposed Model and the Earlier Research

A comprehensive assessment is made at the end of the validation process for the proposed approach. A brief overview is shown in Table 4.10.

Table 4.10. Assessment of the Employed Kaggle and Figshare Datasets with the Methods Currently in Use.

No	Authors	Structure	Year	Data Type	Accuracy (%)
1.	P. Afshar et al. [12]	Capsule Networks	2019	Figshare Dataset-II	90.89
2.	C. L. Choudhury et al. [53]	CNN	2020	Binary Dataset-I	96.08
3.	H. H. Sultan et al. [54]	Resize+ Augmentation + CNN + Hyperparameter Tuning	2019	Figshare Dataset-II	96.13
4.	A. Biswas et al. [55]	Resize+ Anisotropic Diffusion Filter+ Adaptive Histogram Equalization+ DCNN-SVM	2023	Figshare Dataset-II	96
5.	Suhib et. al [5]	Gray Transformation + Resize + Flatten + CNN	2020	Binary Dataset-I	96.7
6.	A. E. Minarno et al. [34]	Resize+ Augmentation + CNN+ Hyperparameter Tuning	2021	Kaggle Dataset-III	96.00
7.	Priyansh et al. [56]	CNN-Based Transfer Learning Approach	2021	Binary Dataset-I	Resnet-50-95, VGG-16- 90, Inception-V3-55
8.	T. Rahman et al. [57]	Resize+ Gray+ Augmentation+ Binary+ CNN	2022	Binary Dataset-I	96.9
9.	H.A. Munira et al. [58]	Thresholding + Cropping+ Resizing+ Rescaling+ CNN-RE and CNN-SVM	2022	Figshare Dataset-II Kaggle Dataset-III	CNN-RF-96.52 CNN-SVM-95.41
10.	T. Rahman et al. [59]	Resize + Gray Transformation + Augmentation + PDCNN	2023	Binary Dataset-I	97.33
				Figshare Dataset-II	97.60
				Kaggle Dataset-III	98.12
11.	**Proposed Method**	**Resize + Gray scale Transformation+ Augmentation + Dilated PDCNN+ Machine Learning Classifiers+ Average Ensemble**	**-**	**Binary Dataset-I**	**98.67**
				Figshare Dataset-II	**98.13**
				Kaggle Dataset-III	**98.35**

Experimental outcomes and evaluation of this proposed dilated PDCNN model alongwith simulation results, performance analysis, and comparative study of this

research are presented in this chapter. Data and output figures with graphical representation are also presenteded.

Chapter 5: CONCLUSIONS

The entire discussion of the method for classifying and detecting brain tumors, as well as its conclusion and future research, will be addressed in this chapter. In Section 5.1, the overall conclusion is explained through a discussion of results, interpretations, and a brief synopsis of the thesis. The difficulties in creating a suggested system that is capable of identifying and categorizing brain tumors are discussed in Section 5.2. Section 5.3 outlines recommendations for future research that include analyzing and modifying other models.

5.1 CONCLUSION

Since brain tumors vary in shape, dimension, and structure, proper identification of these conditions remains extremely difficult. It is well-known how important it is to detect brain tumors early to receive the right medical care. This study proposed a dilated PDCNN structure with machine learning classifiers to detect and classify brain tumors from MRI images. The proposed dilated PDCNN with the average ensemble method is evaluated for binary and multi-class classification on the Kaggle dataset, which contains four different types of tumor images, while the Figshare dataset contains three types of tumor images. The suggested dilated convolution with an expanded receptive field of the kernel has increased the computation efficiency while preserving high accuracy.

With an increasing number of patients, manually analyzing MRI images has grown more complicated, time-consuming, and frequently inaccurate. Conventional machine learning techniques utilize handmade properties, which reduces the solution's durability and raises its cost. Nonetheless, there are situations when supervised learning models perform better than unsupervised learning strategies, leading to an overfitted structure that is inappropriate for

another large database. These problems emphasize how crucial it is to create a fully machine learning-based classification system for brain tumors. By combining the average ensemble technique with PDCNN, this investigation presents a novel approach to the identification and classification of brain tumors. The dilated PDCNN architecture includes both local and global multiscale feature selection paths, a merging phase, and categorization pathways. The initial pictures are converted to grayscale, which makes the process easier. After that, new images are made from old ones by employing data augmentation. Using a modest window size of 5x5 pixels and gradually high dilation rates (4, 2, 1) for each convolution layer, the convolutional layers in the local path collect coarse characteristics and provide local data to the images. In contrast, the global path's convolutional layers obtain fine details by using a large window dimension of 12 by 12 pixels and low dilation rates (2, 1, 1) for every layer of convolution. ReLU activation function and max-pooling layer are applied after each convolutional layer for each path that down-samples the convolutional layer output. A fusion layer connects the two parallel pathways, forming a single path with a cascading link that continues until it reaches the end destination. Two fully connected layers that are attached to a dropout layer that is included in the merging route come after a batch-normalized layer and a ReLU layer. At the output path, the four classifier types—SVM, KNN, NB, and Decision Tree—are used to execute the brain tumor categorization procedure. A regularization method called dropout is also employed to stop the training data from being overfitting.

Among all the performance parameters of dilated PDCNN model with machine learning classifiers on binary Dataset-I, multiclass Figshare Dataset-II and Multiclass Kaggle Dataset-III, including accuracy, precision, recall, and F1-score, binary classification Dataset-I provides the best outcomes. The value of accuracy, precision, recall, and F1-score of dilated PDCNN model on binary classification Dataset-I is 98.67%, 98.62%, 99.17% and 98.28% respectively.

The impact of different dilation rates on the model's accuracy has been examined for the dilated PDCNN. The comparison analysis among various dilation rate arrangements for the various convolution layers demonstrates that the decremented large dilation rate (4, 2, 1) for the local path and the low dilation rate (2, 1, 1) for the global path yield the best results, based on an understanding of the gridding phenomenon and various recommendations for the dilation rate parameter for each layer. For datasets I, II, and III, the highest accuracy values obtained are 98.67%, 98.13%, and 98.35%. This demonstrates that while the global path (lower dilation rates) gains knowledge from the finer features, the local path (higher dilation rates) concentrates on the coarse features. The best outcomes are obtained with this combination.

The results in terms of accuracy, precision, recall, F1-score, error rate, kappa, and training duration, the suggested dilated PDCNN with the average ensemble model performs better than the standard PDCNN with the average ensemble approach. The performance indicators' values will increase even further if the three different types of datasets are dilated to increase the suggested dilated PDCNN's efficiency with the average ensemble model. A thorough comparison is done once the evaluation of the proposed method is accomplished. The findings demonstrate that the suggested simultaneous network topology outperforms detection and classification techniques that have been previously published.

The framework achieved outstanding accuracy, precision, recall, and F1-score regarding the binary brain cancer dataset-I. In order to gain a better understanding of the inner workings of the network and its effectiveness of the dilation rate parameter, experimental evaluation can be performed on other datasets in future investigations. Additionally, studies can be carried out to identify brain tumors with greater accuracy by utilizing actual patient information from any source (various images captured by scanners).

5.2 LIMITATION AND CHALLENGES OF THE STUDY

Owing to their numerous advantages, deep learning algorithms have been applied extensively for the detection and classification of brain tumors. Nevertheless, there are certain restrictions and difficulties with using deep learning algorithms to detect brain tumors, which are detailed below.

- Deep learning algorithms work incredibly well when fed massive amounts of data. For Deep learning algorithms to function properly in brain tumor detection and classification, it is therefore extremely difficult to collect large amount of brain tumor data.
- Deep learning model training is an expensive and time-consuming procedure. Weeks of training in high-end GPU computers are needed for the most sophisticate models. For training the model, a significant amount of memory is also required.
- Again deep learning performs poolyon unbalanced collections of data. The class containing additional information samples is incorrectly classified by it. It is therefore a major issue to have a balanced dataset on brain tumors. Consequently, in cases where the data is unbalanced, deep learning models should be trained using alternative techniques.

5.3 RECOMMENDATION FOR FURTHER STUDY

- This study makes no use of pre-trained models. In this study, a number of pre-trained models like VGG-16, Resnet-50, InceptionV3, AlexNet etc. may be employed on the large dataset.
- This research can be expanded to more datasets, contributing to a better understanding of the performance of brain tumor detection and classification from images.

- Additionally, studies can be carried out to identify brain tumors with greater accuracy by utilizing actual patient information from any source (various images captured by scanners).
- The present study employs four types of classifiers. It is extensible to multiple classifiers.

References

[1] T. A. Sadoon, M. H. A. Al-Hayani., "Deep Learning Model for Glioma, Meningioma and Pituitary Classification," International Journals of Advances in Applied Sciences (IJAAS), vol. 10, no. 1, pp. 88–98, 2021.

[2] M. Kheirollahi, S. Dashti, Z. Khalaj, F. Nazemroaia, P. Mahzouni, "Brain tumors: Special characters for research and banking," Advanced Biomedical Research, 6 January, 2015.

[3] R. Vankdothu, M. A. Hameed, " Brain Tumor MRI Images Identification and Classification Based on the Recurrent Convolutional Neural Network," Measure: Sensors, 28 September, 2022.

[4] [Online]. Available: "Quick Brain Tumor Facts |, Quick Brain Tumor Facts ,"National Brain Tumor Society, 2023.

[5] S. Irsheidat, R. Duwairi, "Brain Tumor Detection Using Artificial Convolutional Neural Networks," 2020 11th International conference on Information and Communication systems (ICICS), pp. 197-203, 2020.

[6] M. Togacar, B.Ergen, Z. Comert , "BrainMRNet: Brain Tumor Detection Using Magnetic Resonance Images with a Novel Convolutional Neural Network Model,"Medical Hypotheses, vol. 134, 1 January 2020.

[7] X. Jiang, Y. Jin, Y. Yao, "Low-dose CT lung images deboising based on multiscale parallel convolution neural network," The Visual Computer, vol. 37, pp. 2419–2431, 2021.

[8] Brain MRI Images for Brain Tumor Detection | Kaggle.

[9] R. Yamashita, M. Nishio, R. K. G. Do, K. Togashi, "Convolutional Neural Networks: an overview and application in radiology," Insights into Imaging, pp. 611-629, 2018.

[10] Z. A. Sejuti, M. S. Islam, "A hybrid CNN-KNN approach for identification of COVID-19 with 5 fold- cross-validation," Sensors Internatinal, vol. 4, pp. 100229, 2023.

[11] L. Guimin, W. qingxiang, Q. Lida, H. Xixian, "Image Super-resolution Using a Dilated Convolutional Neural Networks," Neurocomputing, vol. 275, pp. 1219-1230, 31 January,2018.

[12] P. Afsar, K. N. Plataniotis, A. Mohammadi, "Capsule Networks for Brain Tumor Classification based on MRI Images and Coarse Tumor Boundaries," 2019 IEEE Internatinal Conference on Acoustics, Speech and Signal Processing, Brighton, UK, pp. 1368- 1372, 2019.

[13] Jun Cheng, 2017 Figshare dataset https://figshare.com/articles/brain tumor dataset/1512427.

[14] Brain Tumor MRI Dataset | Kaggle.

[15] A. Anil, A. Raj, H. A. Sarma, N. Chandran et al., "Brain Tumor Detection from Brain MRI using Deep Learning," International Journal of Innovative Research in Applied Sciences and Engineering (IJIRASE)., vol. 3, Issue 2, pp. 68-73,August 2019.

[16] M. Sajjad, S. Khan, K. Muhammad, W. Wu, A. Ullah, S. W. Baik, "Multi-grade brain tumor classification using deep CNN with extansive data augmentation," Journal of Computational Science, vol. 30, pp. 174-182, 2019.

[17] M. A. Habib, "Brain Tumor Detection using Con volutional Neural Networks," Medium, 25 November 2021.

[18] B. Ru, X. Wang, L. Yao, "Evaluation of thr information perspective: determining types of research papers preferred by clinicians," BMC Medical Informatics and Decision Making, vol. 17, pp 5-14, 5 July 2017.

[19] [Online]. Available: "Biopsy: Types of biopsy procedures used to diagnose cancer - Mayo Clinic," [Accessed: 14-February-2024].

[20] [Online]. Available: "CT scan – Mayo Clinic ," [Accessed: 14-February-2024].

[21] [Online]. Available: "Positron emission tomography scan - Mayo Clinic," [Accessed: 14-February-2024].

[22] [Online]. Available: "X-ray - Wikipedia," [Accessed: 14-February-2024].

[23] [Online]. Available: "MRI Basics (case.edu)," [Accessed: 14-February-2024].

[24] D. J. Werring, D. W. Frazer, L.J. Coward, N. A. Losseff, H. Watt et al., "Cognitive dysfunction in patients with cerebral microbleeds on T2-weighted gradient-echo MRI" Brain 127, vol. 127, Issue 10, pp. 2265-2275, October 2004.

[25] J. Hajnal et al., "Use of Fluid Attenuated Inversion Recovery (FLAIR) Pulse Sequences in MRI of the Brain," Journal of Computer Assisted Tomography, pp. 841-844, November 1992.

[26] [Online]. Available: "T1, T2 and flair - Search Images (bing.com)," [Accessed: 15-February-2024].

[27] [Online]. Available: "https://aws.amazon.com/what-is/neural-network," [Accessed: 15-February-2024].

[28] J. M. G. Facundo Bre, Victor D. Fachinotti, "Prediction of wind pressure coefficients on building surfaces using Artificial Neural Networks," Energy and Buildings, vol. 158, pp. 1429-1441, January 2018.

[29] [Online]. Available: "basic structure of ann - Search Images (bing.com)," [Accessed: 15-February-2024].

[30] I. Goodfellow, Y. Bengio, A. Courville, "MIT Press: Cambridge, MA, USA,2016, " Deep Learning. [Accessed: 19-January-2024].

[31] N. Kesav, M. G. Jibukumar, "Efficient and low complex architecture for detection and classification of brain tumor using two channel CNN," Journal of Kinf Saud University- Computer and Information Sciences, vol. 34, pp. 6229-6242, September 2022.

[32] F. J. Diaz-Pernas et al., "A deep learning approach for brain tumor classification and segmentation using a multiscale convolutional neural network," Healthc., vil. 9, no. 2, 2021.

[33] [Online]. Available: "basic structure of CNN with sampling - Search Images (bing.com)," [Accessed: 15-February-2024].

[34] A. E. Minarno, M. H. C.. Mandiri, Y. Munarko, HAriyady, "Convolutional Neural Network with Hyperparameter Tuning for Brain Tumor Classification," Kinetik: Game Technology, Information System, Computer Network, Computoing, Electronics, and Control, vol. 6, no. 2, pp. 127-132, May 2021.

[35] M. K. Abd-Ellah et al., "Parallel deep CNN structure for glioma detection and classification via brain MRI images," 2019 31st International Conference on Microelectronics (ICM), pp. 304-307, 2019.

[36] [Online]. Available: "Convolution operation - Search Images (bing.com)," [Accessed: 15-February-2024].

[37] K. Balaji, K. Lavanya, "Medical image analysis with deep neural networks," Deep Learning and Paralel Computing Environment for Bio-engineering Systems, Academic Press, pp. 75-97, January 2019.

[38] E. Fathi, B. M. Shoja, "Deep neural networks for natural language processing," Handbook of Statistics, vol. 38, p.p. 229-316, Elsevier, 2018.

[39] J. Schmidhuber, “Deep learning in neural networks: An overview,” Neural Networks, vol. 61, pp. 85–117, January 2015, 61.

[40] V. Dumoulin, F. Visin, "A guide to convolution arithmetic for deep learning," arXiv preprint arXiv: 1603.07285, March 2016.

[41] K. Adu, Y. Yu, J. Cai, N. Tashi, "Dilated Capsule Netwrork for Brain Tumor Type Classification via MRI Segmented Tumor Region," International Conference on Robotics and Biomimetics Dali, China, pp. 942-947, December 2019.

[42] S. S. Roy, N. Rodrigues, Y. Taguchi, " Incremental Dilatins using CNN for Brain Tumor Classification," Applied Sciences, vol. 10, Issue 14, pp. 4915, 13 June 2022.

[43] M. A. I. Khan, "Introduction to Softmax Classifier in PyTorch," Deep Learning with PyTorch, January 1 2023. [Accessed: 03-February-2024].

[44] [Online]. Available: "Support Vector Machines. Introduction to margins of separation… | by skilltohire | Medium," [Accessed: 15-February-2024].

[45] J. Cervantes et al., "A comprehensive survey on support vector machine classification: applications, challenges and trends," Neurocomputing, vol. 408, pp. 189-215, September 2020.

[46] C. Zhou et al., "COVID-19 detection based on image regrouping and resnet-SVM using chest x-ray images," IEEE Access, vol. 9, pp. 81902-81912, 2021.

[47] [Online]. Available: "knn classifier with an example - Search Images (bing.com)," [Accessed: 17-February-2024].

[48] M. Yaseliani et al., "Pneumonia Detection Proposing a Hybrid Deep Convolutional Neural Network based on Two Parallel Visual Grometry Group Architectures and Machine Learning Classifiers,” IEEE Access, vol. 10, pp. 62110-62128, 13 June 2022.

[49] S. Y. Sourab, M. A, Kabir, "A comparison of hybrid deep learning models for pneumonia diagnosis from chest radiograms," Sensors International, vol. 3, pp. 10167, 2022.

[50] [Online]. Available: "1.10. Decision Trees — scikit-learn 1.3.1 documentation," [Accessed: 24-October-2023].

[51] [Online]. Available: "Ensemble averaging (machine learning) - Wikipedia," [Accessed: 24-October-2023].

[52] M. Bhuiyan, M. S. Islam, "A new ensemble learning approach to detect malaria from microscopic red blood cell images," Sensors International, vol. 4, pp. 100209, 2023.

[53] C. L. Choudhury, C. Mahanty, R. Kumar, "Brain tumor detection nad classification using convolutional neural network and deep neural network," 2020 International Conference on Computer Science, Engineering and Applicatons (ICCSEA), pp. 1-4, 2020.

[54] H. H. Sultan, N. M. Salim, W. Al-atabany, "Multi-classification of Brain Tumor Images using Deep Neural Network," IEEE Access, pp. 69215-69225, 27 May, 2019.

[55] A. Biswas, M. S. Islam, "A hybrid Deep CNN-SVM Approach for Brain Tumor Classification," Journal of Information Systems Engineering and Business Intelligence, vol. 9, April 2023.

[56] S. Priyansh, A. Maheshwari, S. Maheshwari, "Predictive Modeling of Brain Tumor: A Deep Learning approach," Innovations in Computational Intelligence and Computer Vision, vol. 2, Issue 2, pp. 275-285, 2021.

[57] T. Rahman, M. S. Islam, "MRI Brain Tumor Classification Using Deep Convolutional Neural Network," 2022 3rd International Conference on Innovations in Science, Engineering and Technology (ICISET), 26-27 Februiary, Chittagong, Bangladesh, 2022.

[58] H. A. Munira, M. S. Islam,"Hybrid Deep Learning Models for Multi-classification of Tumour from Brain MRI," Journal of Information Systems Engineering and Business Intelligence, vol. 8, October 2022.

[59] T. Rahman, M. S. Islam, "MRI brain tumor detection and classification using parallel deep convolutional neural networks," Measurement: Sensors, vol. 26, pp. 100694, 2023.