Brain Tumor Detection Using Deep Learning Techniques

By S M Hasan Mahmud

American International University-Bangladesh (AIUB)

Brain Tumor Detection Using Deep Learning Techniques

Showvik Salman Dhrubo (19-39801-1)
Muhtadin Mushfiq (19-40067-1)
Diganto Bhowmik (19-40691-1)
Maizul Islam (19-41002-2)

*A **Thesis** submitted for the degree of **Bachelor of Science (BSc)** in **Computer Science and Engineering (CSE)** at American International University Bangladesh in **May, 2023***

Faculty of Science and Technology (FST)

Declaration by author

Showvik Salman Dhrubo
19-39801-1
Computer Science and Engineering

Muhtadin Mushfiq
19-40067-1
Computer Science and Engineering

Diganto

Diganto Bhowmik
19-40691-1
Computer Science and Engineering

Maizul

Maizul Islam
19-41002-2
Computer Science and Engineering

Approval

Abstract

A brain tumor is an irregular form of cells in the brain or surrounding tissues, and it is one of the deadliest and most severe health problems in the world right now. It can affect people of any age, from children to adults, and early brain tumor detection is necessary for proper treatment and improving patient development. Researchers have employed image processing techniques to identify brain tumors in medical imaging, including MRI, CT, and X-ray scans. Medical images are pre-processed using segmentation, filtering, feature extraction, and classification techniques to identify and locate tumors. In recent years, high accuracy rates have been achieved using Convolutional Neural Networks (CNNs) in brain tumor detection. This thesis proposes a novel method for detecting brain tumors using image processing with various filters, classifiers, and CNN models. The study contains a Binary dataset which is divided into two classes: Brain Tumor and Healthy Brain. Gaussian, Unsharp Mask, and Laplacian filters have been used on these datasets. Three pre-trained models-VGG19, ResNet50, MobileNetV2 and Scratched CNN- have been used. We applied feature extraction using a 1×1024 feature vector, then classified these features into tumor or healthy regions using three classifiers: Random Forest, KNN and AdaBoost. Our results demonstrates that the proposed approach outclasses some existing methods, achieving a classification accuracy of 98.91%. The proposed approach has the ability to aid medical professionals in making more informed decisions and improving patient outcomes.

Keywords

Brain Tumor, MRI, CT, X-ray, CNN, Pre-trained, Feature Extraction, Random Forest, KNN, Filter

Contents

List of Figures

List of Tables

List of Abbreviations and Symbols

Abbreviations	
CNN	Convolutional Neural Network
KNN	K-nearest Neighbor
SVM	Support Vector Machine
SVC	Support Vector Classifier
ResNet	Residual Network
RF	Random Forest
FE	Feature Extraction
etc.	*etc.*

Symbols	
σ	variance and the hyperparameter
etc.	*etc.*

Chapter 1

Introduction

1.1 Thesis Topic

Brain Tumor Detection Using Deep Learning Techniques

1.2 Introduction

Human brain is the most vital organ in the body, controlling the nervous system and other body parts. A brain tumor is one of the most common and dangerous disease of the human brain. There are multiple different kinds of brain tumors. It can be classified as benign and malignant. Malignant tumors are noncancerous and benign tumors are cancerous.[1] Doctors commonly use medical images such as CT scans, magnetic resonance imaging (MRI), and X-rays to diagnose brain tumors. However, according to [2], MRIs are more versatile, and MRIs are better for evaluating soft tissues such as brain tumors than other medical imaging techniques. One of the most frequent forms of cancer are brain tumors, with an estimated 700,000 new cases diagnosed annually. Brain malignancies must be detected early for effective treatment and patient survival.

1.3 Medical Images

Medical images are an essential component of healthcare. They provide visual representations of the human body's internal structures, organs, tissues, and abnormalities. They play a significant role in diagnosing and treatment of various medical diseases. Common types of medical images include magnetic resonance imaging (MRI), X-rays, ultrasound, computed tomography (CT) scans, positron emission tomography (PET), and mammography.

X-ray images use ionizing radiation to detect fractures, tumors, infections, and other conditions primarily in bones and certain organs. CT scans combine multiple X-ray images which is taken from different angles of the body to create detailed images. It also provides information about internal organs, blood vessels, and tissues. MRI uses Strong magnetic fields and radio waves to produce detailed images of organs, soft tissues, and structures of the brain, spinal cord, muscles, and joints. In ultrasound imaging sound waves generate real-time images of internal structures. It is commonly used for heart examinations, abdominal organs, and blood vessels. PET scans inject a small amount of radioactive substance to visualize metabolic activity and detect abnormalities, especially in cancer, cardiovascular disease, and brain tumors. Mammograms consist of X-ray images which are designed for the early detection of breast cancer [3].

These medical images are collected using specialist imaging equipment, and they are analyzed by licensed healthcare professionals to help with diagnosis, planning of the patient's treatment, and monitoring of their progress. Recent advancements in machine learning. Computer vision and Image processing have facilitated automated analysis and interpretation of medical images which helps doctors and physicians in improving accuracy and efficiency during patients' diagnosis and treatment.

1.4 Motivation

Brain tumors are a critical health concern in today's world. Untreated brain tumors can have a devastating effect and have significant consequences for those who have been affected by them. Each year, a substantial number of brain tumor cases are diagnosed worldwide. According to [4], more than 700,000 Americans have a primary brain tumor. Nearly72% of all brain tumors are benign, while only 28% are malignant. In 2023, an estimated 94,390 new cases of primary brain tumors will be diagnosed. As per [5], the tenth leading cause of death is the Brain and other nervous system cancers. In 2023, it is anticipated that 18,990 Americans (11,020 men and 7,970 women) will die from the primary malignant brain and CNS tumors. In 2020, it was estimated that 251,329 people died from primary brain and CNS cancers.
The incidence of brain tumors has increased, and it remains to be an alarming issue in the medical healthcare industry. The existence of a brain tumor in the human brain can significantly affect an individual's quality of life and overall well-being. It is very important to detect and diagnose brain tumors at the earliest stage possible to improve treatment to enhance the survival rate of the affected patient. Early diagnosis and treatment play a vital role in curing brain tumors effectively. Untreated brain tumors have a high risk to the affected, leading to increased healthcare costs and suffering. However, brain tumors can be considerably improved with early detection and the right kind of treatment. Detecting brain tumors early not only enhances the chances of successful treatment but also reduces the overall impact of the disease to the individuals. Therefore, our main motivation is to detect early detection of brain tumors using image processing and machine learning algorithms so that it would be a help to the medical industry.

1.5 Objective

Main objective: The main goal of this paper is to create a model to identify brain tumors from images and improve the model's accuracy.

- **Sub-Objective 1:** To collect the proper dataset for brain tumor and apply Augmentation and filtering using different filters for the improvement of image quality and features.
- **Sub-Objective 2:** To analyze the collected data and train them using our selected Convolution neural Network Models and classify them with our preferred classifiers. we designed this workable method to make it effective and time-saving.
- **Sub-Objective 3:** To explain why we only selected these methods from a wide variety, How the way they operate and how we developed them.

1.6 Orientation

The second chapter of the essay discusses former academicians' research on the same subject, as well as the fundamentals of Machine learning, Image processing and Brain tumor detection.
The third chapter, describes the implementation, data preprocessing, Augmentation, CNN model train and Classifications and a brief overview of the techniques applied to this model.
The Tools and libraries used to build this model and the train-test part are presented in the next section.
The details are wrapped up in Chapter 4 by describing all of the difficulties and comparing the results of all the models to find the best accuracy.
In chapter 5 we discussed the paper and the limitations of our model.
Lastly, in Chapter 6 there are a few concluding observations and future works discussed for our research.

Chapter 2

Literature review

Detection of the brain tumor is one of the most complicated and challenging tasks in medical science because of the complexity of our brains. Doctors and researchers sinuously work to improve brain tumor detection precisely using image processing and deep learning techniques. Using Convolutional Neural Network (CNN) and its model in brain tumor detection is overgrowing because of its high accuracy detection. Researchers are introducing new algorithms and approaches to increase brain tumor detection accuracy.

Deep learning techniques like CNN, have shown great potential in detecting brain tumors in recent years. CNN models can accurately identify regions of brain tumors by learning complex patterns and features from MRI scans. For instance, Dunsheng Liu et al., proposed a new model called Global Average Pooling Residual Network (G-ResNet) for automatic brain tumor detection using (CNN) classification. The model achieves a classification accuracy of 95.00%, significantly better than previous models.[6]

Ouiza Nait Belaid et al., introduced a deep learning technique based on pre-trained VGG-16 CNNs to classify three types of brain tumors using the grey level of co-occurrence matrix (GLCM) features images and original images as inputs. The experimental results demonstrate that when using energy as input, the original image exhibits more prominent and discernible features compared to other combinations of inputs, achieving an average accuracy of 96.5%. [7]
Ramya Mohan and colleagues introduced the MIDNet18 CNN architecture as an alternative to the VGG16 CNN architecture in order to classify standard brain images and brain tumor images. The MIDNet18 model obtained 98.7% accuracy. Whereas the VGG16 model received an accuracy of 50%. [8]

Moreover, researchers have used various CNN models for detecting brain tumors, such as ResNet50, MobileNetV2,and VGG16.For instance, Zahid Rasheed and his team proposed an algorithm that was evaluated using benchmarked data and compared with the performance of pre-trained VGG16, VGG19, ResNet50, MobileNetV2, and InceptionV3 algorithms. The experimental findings revealed a significant classification accuracy of 98.04%. [9]

Additionally, feature extraction has reduced the computational complexity of deep learning models, enabling faster processing and higher accuracy. For instance, Arpit Kumar and colleagues presented a novel approach that incorporates augmentation and feature extraction techniques, employing a machine learning-based ensemble classifier. The proposed method utilizes an optimized fusion vector to accurately identify tumors. This hybrid approach demonstrated outstanding performance, achieving a detection accuracy of 88% when using HOG (Histogram of Oriented Gradients) and a modified ResNet50 model. The results were also compared with the current existing methodology, highlighting the superiority of the proposed approach. [10]

Furthermore
In the classification of extracted features into tumor and non-tumor regions, researchers have explored

various classifiers. Tonmoy Hossain et al., for instance, employed six traditional classifiers, namely SVM, KNN, MLP, Logistic Regression, Naive Bayes, and Random Forest, using the Sci-kit-learn implementation. Subsequently, they experimented with CNN and achieved an impressive accuracy of 97.87%, highlighting its compelling performance.[11]

Aryan Sagar Methil proposed an approach that involved conducting experimental research on a dataset consisting of tumors with diverse sizes, shapes, textures, and locations. The classification task was performed using Convolutional Neural Network (CNN). The CNN model achieved a recall rate of 98.55% on the training set and an impressive 99.73% on the validation set, demonstrating its highly convincing performance.[12]

A. Ari et al., proposed a method that consists of three stages: preprocessing, the extreme learning machine local receptive fields (ELM-LRF) based tumor classification, and image processing based tumor region extraction. In experimental investigations, cranial MR image classification accuracy is 97.18%. The efficacy of the proposed method was superior to that of other recent studies in the literature, according to the evaluation results.[13]

D. Fabbri et al., conducted research on 989 axial images from 191 patients to prevent neural networks from becoming confused by three distinct planes containing the same diagnosis. Classification was accomplished using both completely connected and convolutional neural networks. Additional tests were computed within these two categories by augmenting the original 512*512 axial images. The classification accuracy of neural networks trained on axial data has been demonstrated by an average five-fold cross-validation of 91.43% for the most highly trained neural network. [14]

A. Rehman proposed a new method based on deep learning for detecting and classifying microscopic brain tumors. In the first stage, a 3D CNN architecture is designed to extract brain tumors, and the extracted tumors are transmitted to a CNN model that has been pretrained for feature extraction. Three BraTS datasets from 2015, 2017, and 2018 are used for experiments and validation, and their respective accuracy rates are 98.32, 96.97, and 92.67%. A comparison with extant methods demonstrates that the proposed design achieves comparable precision.[15]

Liya Zhao et al., designed a three-stream framework called multiscale CNNs that could automatically detect the optimal top-three scales of the image sizes and combine information from various scales of the surrounding regions. Multimodal Brain Tumor Image Segmentation Benchmark (BRATS) datasets provided by MICCAI 2013 are used for both training and evaluation. In addition to combining multimodal features from T1, T1-enhanced, T2, and FLAIR MRI images, the designed multiscale CNNs framework combines multimodal features from T1, T1-enhanced, T2, and FLAIR images. Compared to conventional CNNs and the top two methods in BRATS 2012 and 2013 methods, their framework improves the accuracy and robustness of brain tumor segmentation.[16]

A. Deshpande et al., introduced a brain tumor classification framework utilizing artificial intelligence and CNN algorithms. The effectiveness of the framework was evaluated both with and without super-resolution techniques. The combination of super-resolution and the ResNet50 architecture achieved an impressive accuracy of 98.14%. Experimental results using MRI images demonstrated that the proposed super-resolution framework, which incorporates discrete cosine transform (DCT), CNN, and ResNet50, effectively enhances the accuracy of tumor classification. [17]

Imran Javaid et al., demonstrated three distinct neural networks: ResNet-50, CNN and DNN Finally, each simplified neural network is individually allocated to the divided dataset. Once an image is precisely verified as a tumor, OTSU segmentation is used to isolate the tumor. The experimental results demonstrate that the ResNet-50 algorithm has a high classification accuracy of 0.996, precision of 1.00, the highest F1 score of 1.0, and minimal test loss of 0.0269. Extensive experiments demonstrate the effectiveness and precision of the proposed segmentation for tumor detection. [18]

H. Khan et al., proposed a novel approach for categorizing brain MRI scan images as cancerous or noncancerous using a convolutional neural network (CNN) combined with Data Augmentation and Image Processing. They compare their CNN model with pre-trained VGG-16, ResNet-50, and Inception-v3 models using transfer learning. Despite the small dataset used, their model achieves exceptional accuracy of 100%, outperforming VGG-16 (96%), ResNet-50 (89%), and Inception-v3 (75%). Furthermore, their model exhibits low complexity and computational requirements, making it highly efficient and accurate compared to existing pre-trained models. [19]

Ahmet Çinar developed a brain tumor diagnosis system using CNN models and MRI images. They utilized the ResNet50 architecture as the base model, modifying it by removing the last 5 layers and adding 8 new ones. The developed model achieved an impressive accuracy of 97.2%. Additionally, they compared the performance of other models such as AlexNet, ResNet50, DenseNet201, InceptionV3, and GoogLeNet, with the highest performing model accuracy in classifying brain tumor images. The method showcased effectiveness and potential for utilization in computer-aided systems for brain tumor detection, as supported by previous literature studies.[20]

Zhiguan Huang et al., introduced a novel approach called CNNBCN (Convolutional Neural Network based on Complex Networks) for the classification of brain tumors in magnetic resonance imaging (MRI). Unlike manually designed networks, the CNNBCN network structure is generated using randomly generated graph algorithms and then transformed into a computable neural network through a network generator. The modified CNNBCN model achieves an impressive accuracy of 95.49% in brain tumor classification, surpassing other models presented in related studies. Additionally, the model exhibits lower test loss compared to ResNet, DenseNet, and MobileNet models in the conducted experiments. The modified CNNBCN not only delivers satisfactory results in brain tumor classification but also contributes to the methodology of neural network design.[21]

Yakub Bhanothu et al., addresses the time-consuming and error-prone manual evaluation process of MRI images for tumor diagnosis by proposing a deep learning algorithm called Faster R-CNN. The algorithm utilizes the VGG-16 architecture as the base layer for both the region proposal network and the classifier network. It successfully detects tumors and marks their areas of occurrence using the Region Proposal Network (RPN). The algorithm achieves average precision rates of 75.18% for glioma, 89.45% for meningioma, and 68.18% for pituitary tumor. Overall, the algorithm demonstrates a mean average precision of 77.60% across all tumor classes, showcasing its effectiveness as a performance measure in tumor detection and classification. [22]

A. P. Rahmathunneesa et al,. did a research which focuses on evaluating the performance of four pretrained deep learning networks for the classification of brain tumors into four grades. The selected networks, namely AlexNet, GoogLeNet, InceptionV3, and ResNet-50, have been pre-trained on the ImageNet dataset. The study utilizes MRI images of glioma brain tumors and applies skull stripping and data augmentation as preprocessing steps. The network architectures are compared based on metrics such as accuracy, precision, recall, F1-score, and training/validation time. The experimental results indicate that AlexNet achieves an accuracy of 92.98% in a relatively short time of 19 minutes, while ResNet50 achieves a higher accuracy of 96.05% but with a longer duration of approximately 100 minutes. [23]

Tahia Tazin et al., introduced a novel approach focuses on using X-ray images to diagnose brain tumors and improve treatment planning for patients. It explores the application of convolutional neural networks (CNN) for identifying brain tumors, specifically aiming to enhance accuracy through transfer learning. Pretrained CNN models, including VGG19, InceptionV3, and MobileNetV2, were employed for deep feature extraction. The classification accuracy was used to evaluate performance, with MobileNetV2 achieving 92% accuracy, InceptionV3 achieving 91%, and VGG19 achieving 88%. MobileNetV2 demonstrated the highest level of accuracy, enabling early tumor identification before the onset of physical impairments like paralysis. [24]

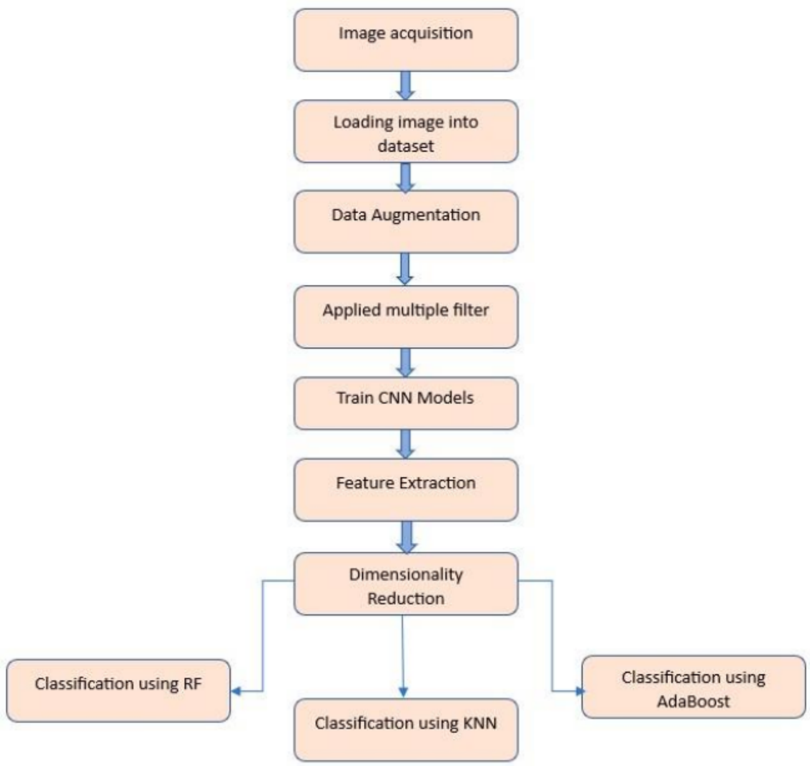
Divjot Kaur et al., proposes a method which highlights the significance of early detection and automated approaches for brain tumor analysis. It introduces three feature extraction models, namely VGG16, VGG19, and Inception v3, which offer effective combination approaches for generating accurate predictions. Machine learning classifiers are employed to categorize brain tumors as benign or malignant. The results indicate that the combination of VGG16 with a neural network classifier achieves the highest accuracy of 99.4%. This research emphasizes the importance of automated methods in improving the accuracy and efficiency of brain tumor diagnosis. [25]

The literature review summarizes that there have been several proposed approaches and techniques for brain tumor detection, including deep learning-based methods, feature extraction, and classifiers using image processing and machine learning techniques. However, there is an area to enhance the precision and effectiveness of these techniques. Using multiple filters, CNN models, feature extraction, and classifiers, This research proposes a more thorough and accurate approach to detecting brain tumors.

Chapter 3

Methods

The proposed methodology includes steps like image acquisition, loading the image into the dataset, data augmentation, gaussian smoothing, image normalization, dimensionality reduction, one hot encoding, and classification. Initially, the images are acquired then the images are pre-processed using augmentation, smoothing, normalizing, etc. Brain tumor images are obtained online containing 2072 images.



18

Fig 3.1 Flow chart for proposed methodology

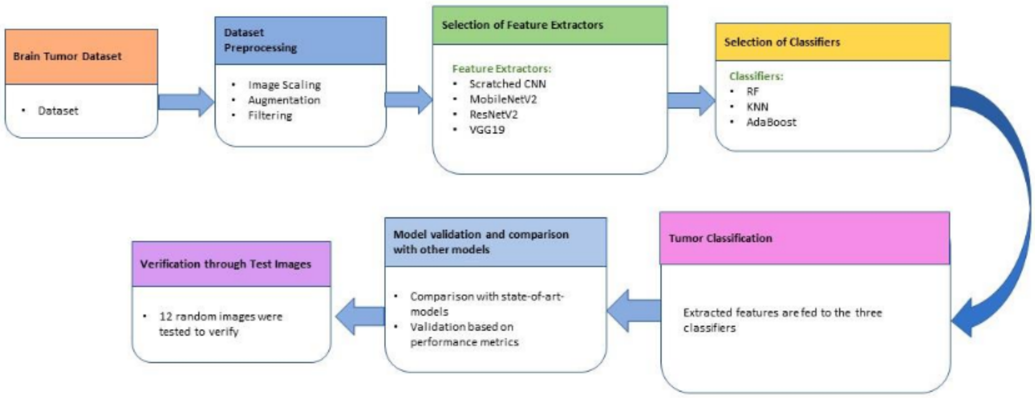
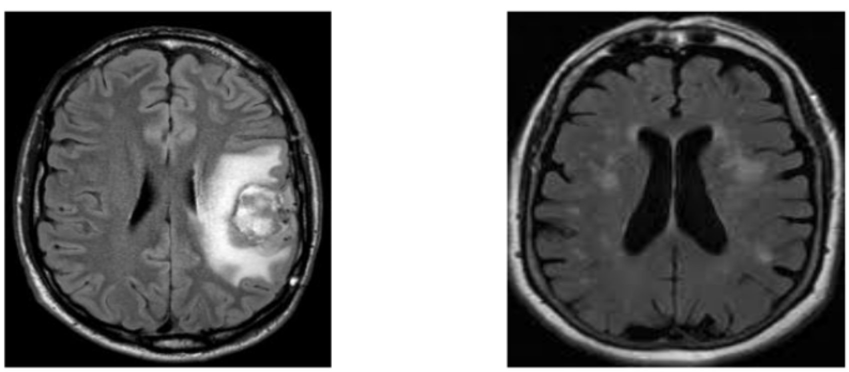


Fig 3.2 Diagram for proposed methodology

3.1 Dataset

A binary dataset [26] is a type of dataset in which each data sample belongs to one of two classes. These classes have binary values of 0 or 1. In this article a Binary dataset was used containing two classes. They are Brain tumor and Healthy brain class. Brain tumor dataset contains 2513 MRi images and healthy brain dataset has 2072 images. Gaussian, unsharp mask [27] and Laplacian [28] filters will be applied on these datasets. These three sets of data will be referred to as Dataset 1, Dataset 2, and Dataset 3, respectively throughout this paper.



(a) Brain Tumor (b) Healthy

Fig 3.3 Dataset classes

3.2 Data preprocessing

Data preprocessing involves cleaning, transforming, and preparing raw data into a format that is suitable for analysis. It includes data transformation, data cleaning, and data normalization. Data preprocessing is important to ensure the accuracy, consistency, and completeness of the data, and to prevent biases or outliers from affecting the analysis.

3.2.1 Filtering

3.2.1.1 Unsharp Mask

Dataset 1 contains MRI images applied with the Unsharp mask filter. Unsharp mask is an image sharpening technique that involves enhancing an image's edges and features by combining a blurred version of the image with the original and then removing the result from the original image. The typical blending formula (1) for unsharp masking is:

$$\text{sharpened} = \text{original} + (\text{original} - \text{blurred}) \times \text{amount} \qquad (1)$$

The unsharp mask function takes an input image as a NumPy array, and two optional parameters radius and amount. radius determines the size of the Gaussian blur kernel, and amount controls the amount of sharpening applied to the image. For the radius value 3 was chosen and value 2 for the amount. The radius parameter controls the size of the kernel used for the Gaussian blur. The Amount applies to the sharpening of an image.

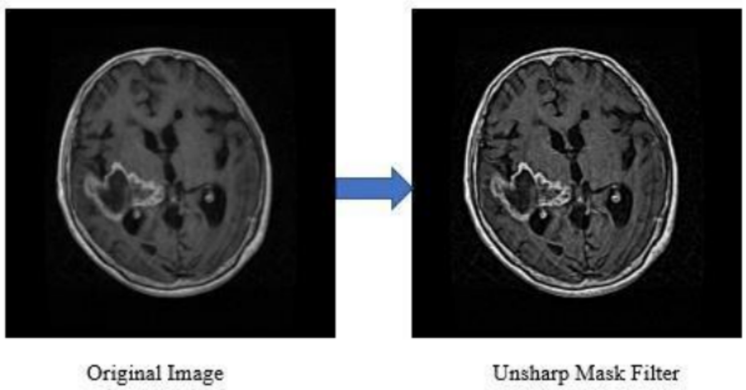


Fig 3.4 Image comparison for Unsharp Mask Filter

3.2.1.2 Gaussian

The Dataset contains Brain Tumor dataset images combined with the Gaussian [29] filter. The gaussian function takes an input image as a NumPy array. It has three parameters. They are sigma, mode and cval respectively. Value 3 is used for sigma and cval value was 0. Mode was constant. Sigma determines the standard deviation of the Gaussian kernel, and mode and cval control how the edges of the image are handled during convolution. The mode and cval parameters control how the edges of the image are handled during convolution.

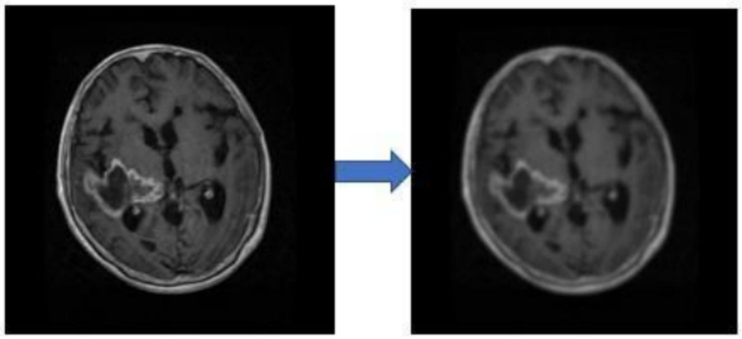


Fig 3.5 Image comparison for Gaussian Filter

3.2.1.3 Laplacian

This dataset contains MRI images of the dataset applied with Laplacian filter. The Laplacian is an image processing operator that is used for edge detection, image sharpening, and other image enhancement tasks. This creates a 3x3 Laplacian kernel, which is used for filtering the image. This applies the Laplacian kernel to the input image using the cv2.filter2D function from the OpenCV [30] library. There are three parameters available here which are src, ddepth and kernel. Src specifies the input image. Ddepth means the depth of the output image and kernel specifies the Laplacian kernel. The pixel values of the generated image are clipped to make sure they are between [0, 255]. The amount of amplification is controlled by the C variable and gClip variable is the output of unsharp mask operation.

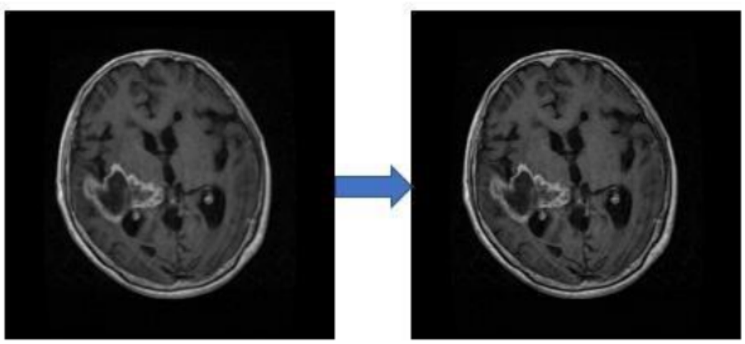


Fig 3.6 Image comparison for Laplacian Filter

3.2.2 Data Augmentation

Minimum preprocessing is done on the datasets - image resizing and augmentation. All the images are resized to 224* 224 pixels. First the dataset was taken as input image data. Then the rescale parameter is used to normalize the pixel values of the images by dividing them by 255 [31]. Then the dataset was splitted into validation and training sets using validation_split (2) which was set to 20%.

$$\text{Validation Split} = \frac{Validation\ size}{Total\ size} \qquad (2)$$

also, the zoom range (3) was set to 0.99 to randomly zoom in or out on the images. After that the image data is passed to the preprocessing train function which takes a path to a directory containing the training images and returns an image generator object.

$$zoom\_range = (min\_zoom, max\_zoom) \qquad (3)$$

After that image_data object is used to generate batches of training images. The target size is set to 224*224 pixels.each batch will contain 8 images [31] . And the seed has a value of 123 to ensure reproducibility of the results. Finally, the subset parameter is set to 'training' to indicate that this is the training set.

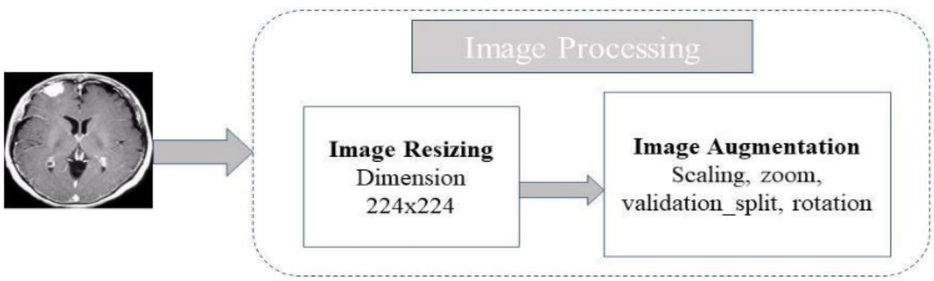


Fig 3.7 Data Preprocessing stages

3.3 Feature Extraction

Feature extraction was applied on the dataset as it helps to transform raw data into a format that can be used to build accurate, efficient, and interpretable models. Extracting the exact tumor is a crucial task in case of brain tumor because of the complex structure of the brain. Certain parameters are taken into account for feature extraction as size, shape, composition, location of image. As per the result obtained from the feature extraction the classification of the tumor is done. For feature extraction, three pre-trained models-VGG16, Resnet50, Mobilenetv2 and a Scratch CNN model from were considered. We have applied all these pretrained models on the dataset that are mentioned above. Feature extraction was applied on the trained data where we took 1*1024 features from each image.

3.3.1 Modified Scratched CNN

Convolutional Neural Networks [32] are a type of neural network designed to process data that has a grid-like structure, such as images or audio spectrograms. CNN learns and extracts features from input images through a series of convolutional filters [33], activation, pooling, and fully-connected layers. In this study, the CNN architecture has been tuned and refined by making some modifications to enhance its performance. This modified CNN architecture consists of several layers, including convolutional layer (4), max pooling layers, a flatten layer, and dense layers.

$$z_i = b_i + \sum_{j=1}^{F} \sum_{k=1}^{F} w_{j,k} x_{i+j-1,i+k-1} \qquad (4)$$

The first layer is an input layer. It takes an image of shape (224, 224, 3). Here 224*224 represents height and width and 3 represents the RGB color channels. The input layer has three convolution layers with 32 and 64 filters using a 3x3 kernel size and ReLU activation [34] function. Then max pooling (5) layers are added to reduce the spatial dimensions of the feature.

$$y_{i,j} = \max_{u=i\times W}^{(i+1)\times W-1} \max_{v=j\times H}^{(j+1)\times H-1} z_{u,v} \qquad (5)$$

The output of the last max pooling layer is then flattened into a 1D array by the Flatten layer. After that flattened layer is passed through the dense layers with 2048 and 1024 units, respectively. To improve the training process a dropout layer is added after each dense layer to prevent overfitting. The dropout layer rate is 0.3. Lastly the output layer consists of two units along with a sigmoid activation (6) function. [35]

$$f(x) = 1 / (1 + \exp(-x)) \qquad (6)$$

This modified CNN architecture with dropout and batch normalization techniques improves the model's accuracy and stability.

Layer (type)	Output Shape	Param
conv2d (Conv2D)	(None, 224, 224, 32)	896
max_pooling2d (MaxPooling2D)	(None, 112, 112, 32)	0
conv2d_1 (Conv2D)	(None, 112, 112, 64)	18496
max_pooling2d_1 (MaxPooling2D)	(None, 56, 56, 64)	0
flatten (Flatten)	(None, 200704)	0
dense (Dense)	(None, 1024)	205521920
batch_normalization (BatchNo)	(None, 1024)	4096
dropout (Dropout)	(None, 1024)	0
dense_1 (Dense)	(None, 2)	2050

Table 3.1 Summary of the proposed Scratched CNN based on two-class classification.

3.3.2 Modified MobilenetV2

MobileNetV2 [37] is a variant of the MobileNet architecture, which is a type of convolutional neural network designed for mobile and embedded devices with limited computational resources. It was introduced by Google in 2018.The input shape is set to (224, 224, 3), and the pre-trained layers are set to non-trainable. A dropout layer with a rate of 0.3 is added to prevent overfitting, and a dense layer with 3 units and sigmoid activation [38] function is added. The model is compiled using the Adam optimizer, sparse categorical cross-entropy (8) as the loss function, and accuracy metric for evaluation.

$$Loss = -\sum_{i=1}^{n} y_i * \log \hat{y}_l \qquad (7)$$

These modifications make the model faster and provide better accuracy.

Layer (type)	Output Shape	Param
mobilenetv2_1.00_224	(None, 1280)	2257984
flatten (Flatten)	(None, 1280)	0
dense (Dense)	(None, 1024)	1311744
dense_1 (Dense)	(None, 2)	2050

Table 3.2 Summary of the proposed MobileNetV2 based on two-class classification.

3.3.3 Modified Resnet50

ResNet50 [36] is a model of the Residual Network architecture, which is a type of neural network designed to address the problem of vanishing gradients in deep neural networks. ResNet50 was introduced in 2015 by Kaiming. He and his colleagues at Microsoft Research. It is a 50-layer deep neural network that consists of a series of convolutional layers.
The original model is trained on a large-scale image classification dataset with over 1000 categories. This modified ResNet50 model has fewer trainable parameters than the original ResNet50 model, which makes it faster to train and requires less computational resources.

$$H_p(q) = -\frac{1}{N}\sum_{i=1}^{N} z_i\log\big(p(z_i)\big) + (1 - z_i)\log\big(1 - p(z_i)\big) \quad (8)$$

Also, this model uses binary cross-entropy (7) [36] as the loss function and the Adam optimizer with a learning rate of 0.001, while the original model may use different loss functions and optimizers depending on the training task.

Layer (type)	Output Shape	Param
resnet50 (Functional)	(None, 2048)	23587712
flatten (Flatten)	(None, 2048)	0
dense (Dense)	(None, 1024)	2098176
dense_1 (Dense)	(None, 2)	2050

Table 3.3 Summary of the proposed ResNet50 based on two-class classification.

3.3.4 Modified VGG19

VGG19 [39] is a deep convolutional neural network architecture that was proposed by researchers at the University of Oxford in 2014. VGG19 [39] consists of 19 layers. 16 of them are convolutional layers and 3 are fully connected layers. Here VGG19 architecture has been fine-tuned with some modification. Transfer learning has been used in this modification whereas a pre-trained VGG19 model with ImageNet weights is used as the base model. The last convolutional block and the dense layers of the original VGG19 model are removed, and two new dense layers are added with 1024 and 2 units, respectively. The output of the pre-trained VGG19 model is flattened and passed through the new dense layers. The learning rate of the Adam optimizer is set to 0.001. categorical cross entropy (9) was used as the loss function and accuracy (7) was set as the evaluation metric.

$$H(p, q) = -\sum_x p(x)\log q(x) \quad (9)$$

This to avoid overfitting and improve the training efficiency.

Layer (type)	Output Shape	Param
vgg19 (Functional)	(None, 512)	20024384
flatten_1 (Flatten)	(None, 512)	0
dense_2 (Dense)	(None, 1024)	525312
dense_3 (Dense)	(None, 2)	2050

Table 3.4 Summary of the proposed VGG19 based on two-class classification.

3.4 Classifier

3.4.1 Random Forest

Random Forest [40] is used for classification and regression tasks. It combines multiple decision trees to make more accurate predictions. The algorithm randomly selects a subset of the features from the dataset and then constructs multiple decision trees based on these features.
The final prediction is made by aggregating the predictions of all the trees in the forest, typically using a majority vote for classification or a mean value for regression. In random forest classifier n_estimators have been set to 100 so it will create 100 decision trees and random seed value is set to 4020 so the same results can be reproduced if the code is run again with the same random seed. In random forest the most frequently predicted class is voting (10) The equation for margin function (11) for random forest:

$$f(x) = \arg \max_{y \in \mathcal{Y}} \sum_{j=1}^{J} I\big(y = h_j(x)\big) \qquad (10)$$

$$mg(\mathbf{X}, Y) = av_k I(h_k(\mathbf{X}) = Y) - \max_{j \neq Y} av_k I(h_k(\mathbf{X}) = j) \quad (11)$$

3.4.2 K-Nearest Neighbors (KNN)

KNN [41] works by finding the k nearest data in the training set to a given test point, and then it uses the class or average value of these neighbors to predict the test point. A smaller value of K results in a more intricate model with lower bias and higher variance. On the other hand a larger value of K yields a simpler model with higher bias and lower variance. The optimal value of K is typically determined through model selection techniques, such as cross-validation.Here the number of neighbors has been set to 2 so two nearest neighbors will be considered for each prediction. Minkowski metric specifies the distance metric to use for calculating the distance between points in the dataset. The Minkowski distance metric (12) will be the same as the Euclidean distance metric. The equation of distant function (13) and h(x) (14) for KNN:

$$\mathrm{dist}(\mathbf{x}, \mathbf{z}) = \left(\sum_{r=1}^{d} |x_r - z_r|^p\right)^{1/p} \qquad (12)$$

$$\mathrm{dist}(\mathbf{x}, \mathbf{x}') \geq \max_{(\mathbf{x}'', y'') \in S_{\mathbf{x}}} \mathrm{dist}(\mathbf{x}, \mathbf{x}'') \qquad (13)$$

$$h(\mathbf{x}) = \text{mode}(\{y'' : (\mathbf{x}'', y'') \in S_{\mathbf{x}}\}) \qquad (14)$$

3.4.3 AdaBoost

AdaBoost [42] classifier works by combining multiple weak classifiers into a strong classifier, with each weak classifier focusing on different aspects of the data. AdaBoost iteratively train a series of weak classifiers on the training data, with each subsequent classifier focusing more on the examples which was misclassified by the previous classifier. During each iteration, AdaBoost assigns weights to the training examples, with more weight given to the examples that were misclassified by the previous classifiers. The final prediction is then made by combining the predictions of all the weak classifiers, weighted by their accuracy. It iteratively train a sequence of weak classifiers on the training data [42], where each subsequent weak classifier places more emphasis on the data points that were misclassified by the previous weak classifier. Here the value of n_estimator parameter is set to 10 so that it will use 10 weak classifiers to boost the performance of the model and the random seed value is set to 2020 so the same results can be reproduced if the code is run again with the same random seed. The sign function (15) and weighted error rate for weak classifier (16) and loss (17) function for AdaBoost classifier:

$$H(x) = \text{sign}(\textstyle\sum_{t=1}^{T} \alpha_t h_t(x)) \qquad (15)$$

$$\epsilon_m = \textstyle\sum_{y_i \neq k_m(x_i)} w_i^{(m)} / \sum_{i=1}^{N} w_i^{(m)} \qquad (16)$$

$$\varepsilon_m = \frac{\sum_{n=1}^{N} w_n^m \ell(f_m(\mathbf{x}_n), y_n)}{\sum_{n=1}^{N} w_n^m} \qquad (17)$$

3.5 Tools and Libraries Used

3.5.1 TensorFlow

TensorFlow [43] is a powerful and popular software library combined with artificial intelligence and machine learning. It is open source and freely available. The development of TensorFlow is led by the Google Brain team. It is primarily implemented in C++ but provides a Python interface for easier usage and interaction with the C++ framework.

Python serves as a convenient interface for users to interact with TensorFlow. Python is also used for computation, machine learning, and deep learning tasks. To facilitate a wide range of tasks, TensorFlow uses symbolic math computations, data flow, and differentiable programming, with a focus on deep neural network training. TensorFlow code is written in Python. However, when developing new methods or features C++ is often employed.

TensorFlow is utilized in various deep learning applications. It offers pre-built architectures for deep learning and machine learning algorithms which includes convolutional neural networks for computer vision, natural language processing, Image processing and neural network for tasks like sequence labeling, classification, and prediction. [43] It is an open-source platform for machine learning that provides a comprehensive support from training to deployment.

TensorFlow's capabilities include a wide range of fields such as handwritten digit classification, image recognition, word embedding, recurrent neural networks, machine translation using sequence models, natural language processing and others. TensorFlow enables a smooth transition between training and large-scale production by using the same models.

It is a versatile framework that empowers researchers and Scientists to tackle complex machine learning tasks, functionalities and ease of use through Python interface.

3.5.2 Kaggle

Kaggle is a widely recognized platform that gives easy access to the materials and datasets in the field of Machine learning and data science. It is particularly popular for deep learning tasks like neural networks. Kaggle offers a range of functionalities to the users to discover and explore a wide range of datasets and construct models within a website. It also allows to collaborate with other data scientists and machine learning engineers to overcome the challenges within this domain. Competitions are often hosted in the website where the users and participants can showcase their skills.
Kaggle's dataset helps us by providing valuable materials and resources which creates an active community. Whether you are a beginner trying to learn new skills and establish a project or an experienced individual seeking to participate in competitive activities, Kaggle is the ideal destination. The platform offers a wide number of topics, including Python programming, machine learning, data visualization, SQL, deep learning, natural language processing (NLP), and Image processing. It provides in-depth explanations and assistance for these topics which makes it a helpful and informative resource for anybody with an interest in this domain.

3.5.3 Scikit Learn

Scikit-learn [44] is a highly versatile and robust machine learning library widely used in Python. It builds upon popular Python packages such as NumPy and SciPy. It offers a comprehensive set of algorithms for classification, regression, clustering, supervised, and unsupervised machine learning tasks. It is based on python interface which gives access to a range of efficient tools for machine learning that includes classification, regression, clustering, and dimensionality reduction techniques. NumPy, SciPy, and Matplotlib are some well-known Python libraries that Scikit-learn simply interacts with. It has a collection of machine-learning algorithms, which includes logistic regression, support vector machines (SVM), and random forest. These algorithms enable data scientists to perform classification, regression, and clustering easily.
Users can access a rich variety of functionality for exploring and analyzing data, feature engineering, model training, and evaluation. The library is quite user-friendly. It offers a consistent, user-friendly interface, making it appropriate for both beginner and experienced data scientists and machine learning practitioners. Scikit-learn is a popular and widely used machine learning package in Python which is known for its usability and robustness. It helps the user to tackle various machine learning tasks efficiently and effectively.

3.5.4 OpenCV

OpenCV [45] is a powerful and widely used library for real-time computer vision, machine learning, and image processing tasks. It plays a crucial role in the sector of image processing. OpenCV provides an extensive range of tools and functions for image processing and computer vision. With OpenCV we can process images and videos to detect objects and faces.
This library has a versatile set of functionalities, including face detection, object tracking, landmark detection, and much more. It supports multiple programming languages, including Python, Java, and C++.OpenCV provides a enrich library content that is for researchers and students in the fields of computer vision and image processing.

It is widely used for analyzing and processing images. It enables tasks such as rotating images at arbitrary angles from the center, downscaling images, and applying filters like Gaussian blur to smoothen the images which improves computational efficiency. The library provides efficient and optimized algorithms that enable real-time processing of image data. By using OpenCV, developers and researchers can implement advanced computer vision techniques and algorithms with ease. It is widely used in object recognition, augmented reality, robotics, and surveillance systems.

3.5.5 Matplotlib

Matplotlib [46] is a popular and powerful data visualization library in Python that has a wide range of high-quality plots and charts. It provides a flexible and user-friendly interface for creating several types of visualizations that effectively communicate data insights.
With Matplotlib, we can create line plots, scatter plots, bar plots, histograms, pie charts, and more. It has a set of customization options, allowing you to fine-tune every aspect of plots. Colors, line styles, markers, axis labels, titles, legends, and other visual elements can be customized for user's specific needs. It integrates with other Python libraries, such as NumPy and Pandas, making it easy to visualize data arrays and perform data analysis tasks. It can also integrate with Jupyter notebook, allowing for interactive plotting and exploration.
Matplotlib is widely used in various domains, including data science, machine learning, scientific research, and data visualization. It enables user to visualize patterns, trends, distributions, and relationships within the data. It helps data exploration, analysis, and presentation. The library supports different output formats that includes interactive plots and static images for publications. It also provides capabilities for creating animated visualizations, which can be useful for presentations or web applications. Matplotlib is a versatile and essential tool for creating visually appealing and informative plots and charts in Python. Its wide range of plot types, customization options, and integration make it a valuable resource for data visualization and analysis tasks.

3.6 Train Test Split

In deep learning model training, it is common practice to divide the dataset into separate sections: the training section, testing section, and validation section. Typically, when we obtain a primary dataset, it comes as a single directory or file. To work with the dataset effectively, we need to preprocess it and divide it into suitable subsets for training and testing.In many cases, dividing the dataset into training and testing subsets is essential, and the scikit-learn library provides a convenient function for this purpose. This function is commonly used to split arrays or datasets into different subsets for training and testing. The function in scikit-learn takes several parameters, including the arrays or datasets to be split, the desired test size and randomization options. By calling this function and specifying the appropriate parameters, the dataset can be randomly divided into training and testing subsets.This splitting process ensures that the model is trained on a portion of the data and then evaluated on unseen data to assess its performance. The validation section is often considered separately and can be created from the training data or used in combination with the testing data to fine-tune and validate the model.By following this approach, we can effectively prepare the dataset for deep learning model training, ensuring that the model is trained on a representative portion of the data and evaluated on independent data to assess its generalization capabilities.
We used 80% data for training. There were 2348 items in our training subset. We have allocated 20% data for testing. the test subsets contained around 587 items.When the primary data is not initially divided into training and testing sections, it is indeed a common mistake to train models using the

entire dataset, which often leads to overfitting. Overfitting occurs when the model becomes too specialized in learning the training data and fails to generalize well to unseen data.
To avoid this issue, it is crucial to split the data into distinct subsets for training, testing, and validation.
The training data is used to train the model, the testing data is used to evaluate the model's performance on unseen data, and the validation data is used to fine-tune the model and make decisions regarding hyperparameter tuning or model selection.
It is important to emphasize that tampering with the testing data is highly unethical and can lead to biased and misleading results. The testing data should be kept separate and not be used in any way during model development.

3.7 Evaluation Metrics

3.7.1 Precision

Precision metric is used to evaluate the performance of a classification model. It is calculated by dividing the total number of true positive samples by the sum of true positive and false positive samples. Precision measures how accurately the model identifies positive samples among the samples it predicted as positive.
Precision can also refer to the level of accuracy or closeness between two or more measurements. It quantifies the degree of similarity or agreement between different measurements or values. Precision can be described as the level of exactness or accuracy with which a particular dimension or quantity is determined or measured. It indicates how closely related or similar two or more measures are to each other, reflecting the level of accuracy in the measurement process. The equation (18) is

$$\text{Precision} = \frac{Tp}{Tp+Fp} \qquad (18)$$

3.7.2 Accuracy

Accuracy metric is a commonly used metric for evaluating the performance of classification models. However, accuracy may not be the preferred metric in many cases, especially when dealing with datasets that have imbalanced classes. In such cases, a classifier that predicts the majority class for every instance can achieve a high accuracy, even though it fails to capture the minority class. In our particular dataset, we have multiple classes, and the accuracy metric may not provide the desired results due to class imbalances. The equation (19) is

$$\text{Accuracy} = \frac{Tp+Tn}{Tp+Tn+Fp+Fn} \qquad (19)$$

3.7.3 Recall

Recall is an important evaluation metric, especially in scenarios where identifying positive instances correctly is crucial. It is also known as sensitivity or true positive rate. Recall measures the ability of a classifier to correctly identify all positive instances out of the total actual positive instances present in a dataset. True positives (TP) represent the instances that are correctly classified as positive by the classifier. False negatives are the instances that are actually positive but are incorrectly classified as negative by the classifier. The equation (20) is

$$\text{Recall} = \frac{(Number\ of\ true\ positives)}{(Number\ of\ true\ positives + Number\ of\ false\ negatives)} \qquad (20)$$

3.7.4 F1 Score

The F1 score is a metric that combines both precision and recall to provide a balanced evaluation of a classifier's performance. It is particularly useful when dealing with imbalanced datasets or when there is an uneven cost associated with false positives and false negatives.
It provides a single metric that balances both precision and recall. It ranges between 0 and 1, where 1 represents the best possible F1 score and 0 indicates the worst F1 score.
The F1 score (21) is the harmonic mean of precision and recall and can be calculated using the following equation:

$$\text{F1 Score} = \frac{2 * (Precision * Recall)}{(Precision + Recall)} \qquad (21)$$

Chapter 4

Results or findings

4.1 Analysis of CNN Models for Original Dataset

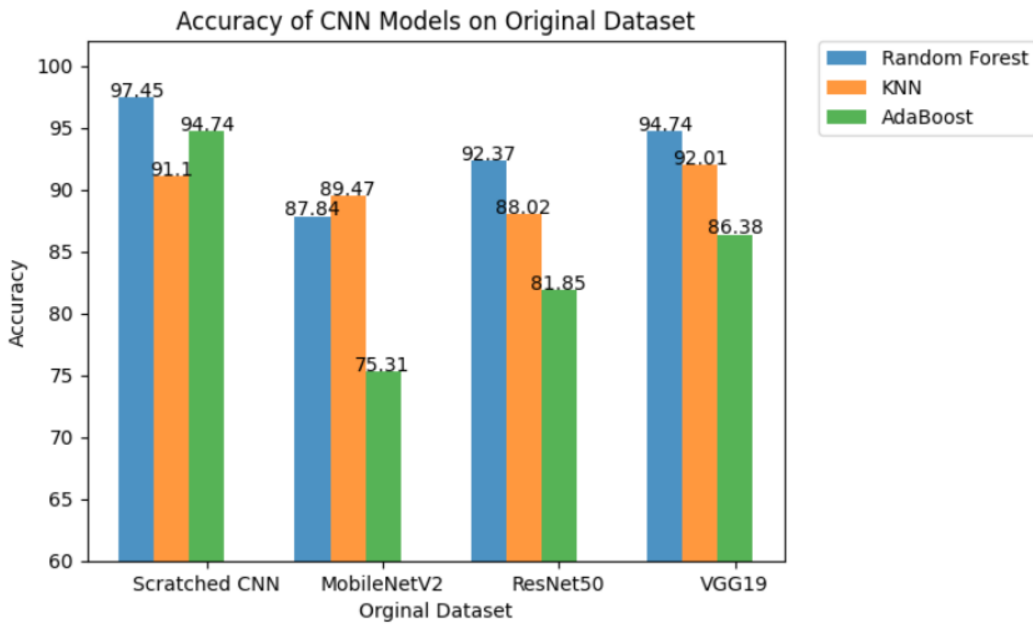


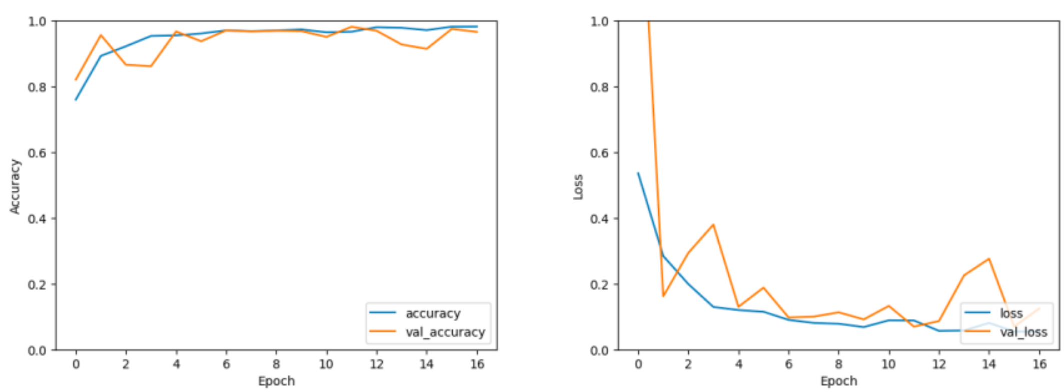
Fig 4.1 Histogram of the Accuracy of CNN Models on Original Dataset

The histogram Fig 4.1 delineates the accuracy of various CNN models on the original dataset. The comparisons are carried out between Random Forest, KNN and AdaBoost classifiers in accordance with the model training implementing these classifiers. Scratched CNN using Random Forest has the greatest accuracy within all the models which is 97.45%. The accuracy of the Random Forest classifier is on the higher accuracy range compared to the other two classifiers having 92.37% and 94.74% when ResNet50 and VGG19 models were implemented. The accuracy of Random Forest is lowest in MobileNetV2 which is 87.84%. AdaBoost performs much worse than the other two classifiers in three of the models, MobileNetV2, ResNet50, and VGG19 with accuracy of 75.31%, 81.85%, and 86.38%, respectively. However, the accuracy of the AdaBoost classifier rose to 94.74% while using the Scratched CNN model. The KNN classifier is on the middle of the range in terms of accuracy for two of the models which are ResNet50 and VGG19 whereas it showed substantial growth when MobileNetV2 model has been used as it had the highest accuracy among all three classifiers which is 89.47%. Overall, it can be inferred that the use of different classifiers along with which model has been used to process a dataset have a compelling impact on the degrees of accuracy while using the same dataset.

4.1.1 Performance Results

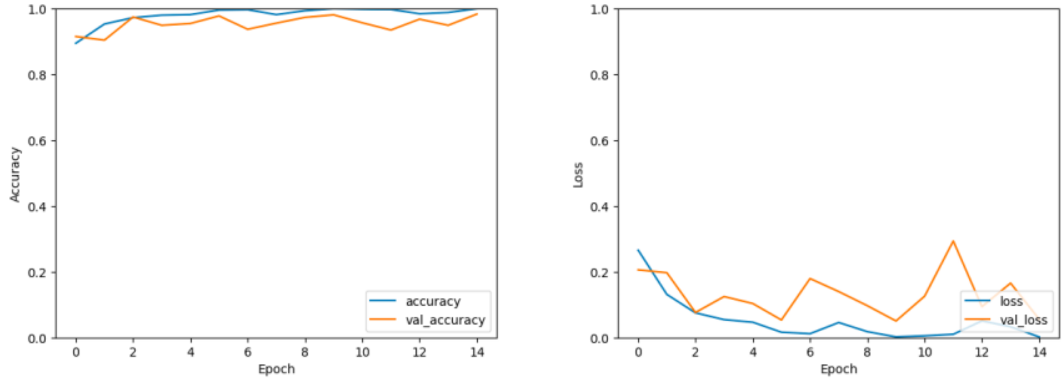
Model	Accuracy	Loss	Validation Accuracy	Validation Loss
Scratched CNN	0.9819	0.0541	0.9659	0.1247
MobileNetV2	1.0000	0.0018	0.9835	0.0553
ResNet50	0.8314	0.3596	0.8460	0.3757
VGG19	0.9666	0.0941	0.9681	0.0909

Table 4.1 Accuracy Results for Original Datasets.



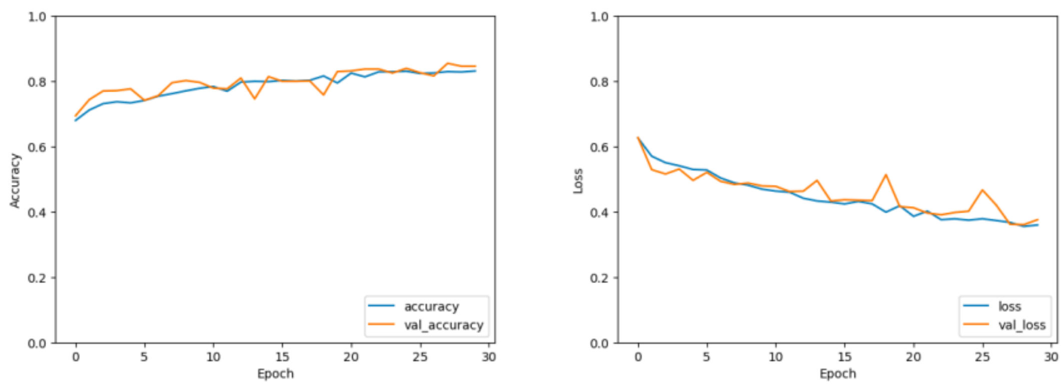
a. Accuracy (b) Loss

Fig 4.2 Performance Results of Scratched CNN.



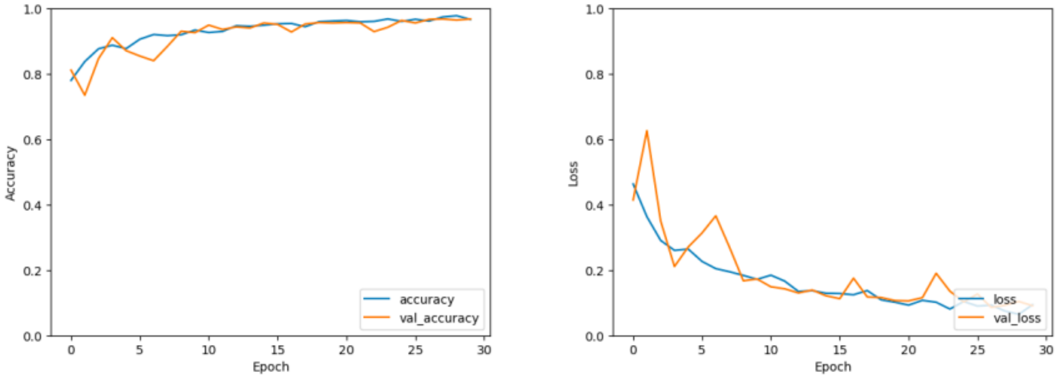
(a) Accuracy (b) Loss

Fig 4.3 Performance Results of MobileNetV2.



(a) Accuracy (b) Loss

Fig 4.4 Performance Results of ResNet50.



(a) Accuracy (b) Loss

Fig 4.5 Performance Results of VGG19.

4.1.2 Performance Metrics

Models	Class	Precision	Recall	F1-Score
Scratched CNN	Brain Tumor	96.15%	99.33%	97.19%
	Healthy	99.16%	95.18%	97.13%
MobileNetV2	Brain Tumor	89.67%	88.81%	89.24%
	Healthy	85.47%	86.55%	86.01%
ResNet50	Brain Tumor	93.63%	93.04%	93.33%
	Healthy	90.72%	91.49%	91.10%
VGG19	Brain Tumor	93.73%	96.59%	95.15%
	Healthy	95.97%	92.60%	94.25%

Table 4.2 Precision, Recall and F-1 Score for Random Forest Classifier

Models	Class	Precision	Recall	F1-Score
Scratched CNN	Brain Tumor	86.25%	99.66%	92.47%
	Healthy	99.50%	80.72%	89.13%
MobileNetV2	Brain Tumor	86.96%	95.85%	91.18%
	Healthy	93.68%	81.09%	86.94%
ResNet50	Brain Tumor	93.10%	83.44%	89.10%
	Healthy	82.37%	91.49%	86.69%
VGG19	Brain Tumor	89.55%	96.25%	92.78%
	Healthy	95.32%	87.16%	91.03%

Table 4.3 Precision, Recall and F-1 Score for KNN Classifier

Models	Class	Precision	Recall	F1-Score
Scratched CNN	Brain Tumor	93.61%	97.02%	95.28%
	Healthy	96.22%	91.96%	94.04%
MobileNetV2	Brain Tumor	79.59%	76.04%	77.77%
	Healthy	70.24%	74.37%	72.37%
ResNet50	Brain Tumor	85.52%	82.27%	83.87%
	Healthy	77.33%	81.27%	79.25%
VGG19	Brain Tumor	85.44%	89.79%	87.56%
	Healthy	87.60%	82.90%	84.97%

Table 4.4 Precision, Recall and F-1 Score for AdaBoost Classifier

4.2 Analysis of CNN Models for Unsharp Mask Filter

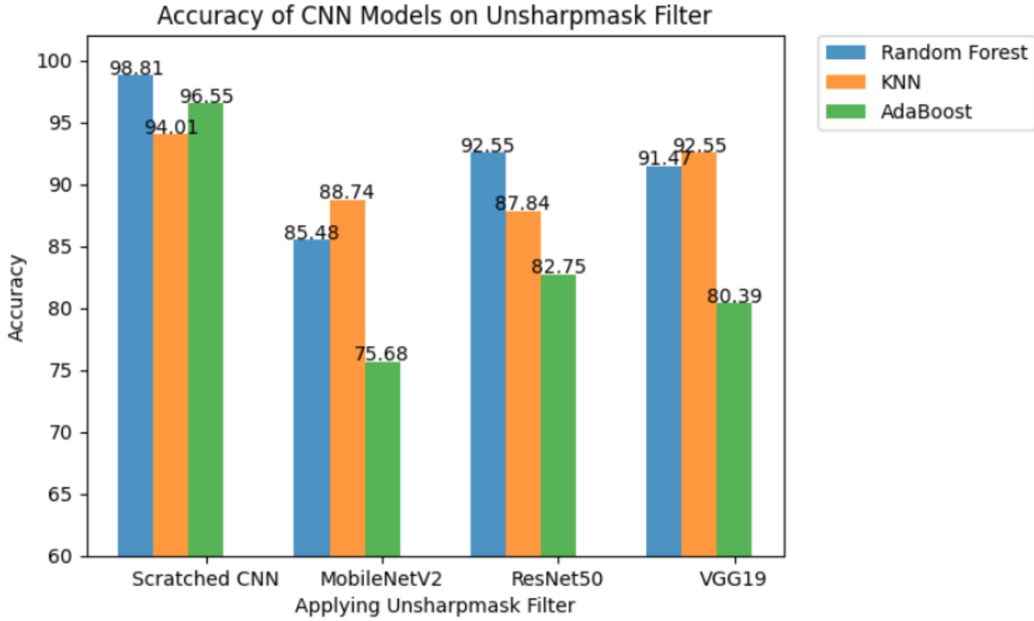


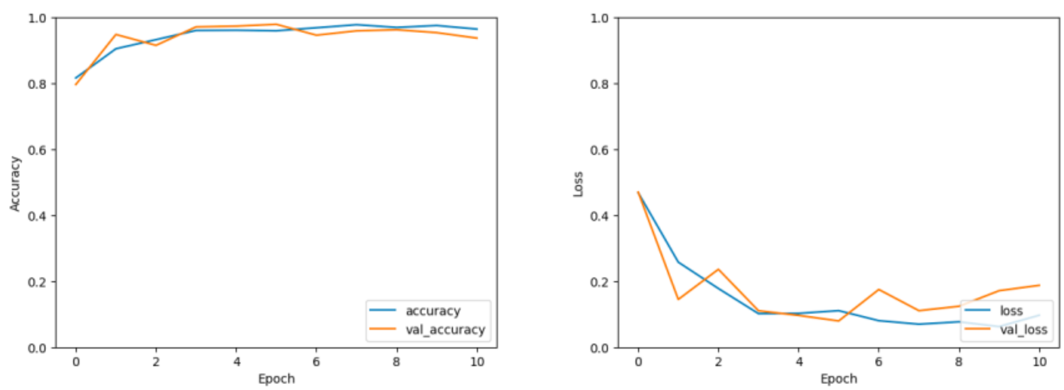
Fig 4.6 Histogram of the Accuracy of CNN Models on Unsharp Mask Filter

The histogram Fig 4.6 delineates the accuracy of various CNN models on the Unsharp Mask filter. The comparisons are carried out between Random Forest, KNN and AdaBoost classifiers in accordance with the model training implementing these classifiers. Scratched CNN using Random Forest has the greatest accuracy within all the models which is 98.81%. The accuracy of the Random Forest classifier is on the higher accuracy range compared to the other two classifiers having 92.55% and 94.47% when ResNet50 and VGG19 models were implemented. The accuracy of Random Forest is lowest in MobileNetV2 which is 85.48%. AdaBoost performs much worse than the other two classifiers in three of the models, MobileNetV2, ResNet50, and VGG19 with accuracy of 75.68%, 82.75%, and 80.39%, respectively. However, the accuracy of the AdaBoost classifier rose to 96.55% while using the Scratched CNN model. The KNN classifier is on the middle of the range in terms of accuracy for two of the models which are ResNet50 and VGG19 whereas it showed substantial growth when MobileNetV2 model has been used as it had the highest accuracy among all three classifiers which is 88.74%. Overall, it can be inferred that the use of different classifiers along with which model has been used to process a dataset have a compelling impact on the degrees of accuracy while using the same dataset.

4.2.1 Performance Results

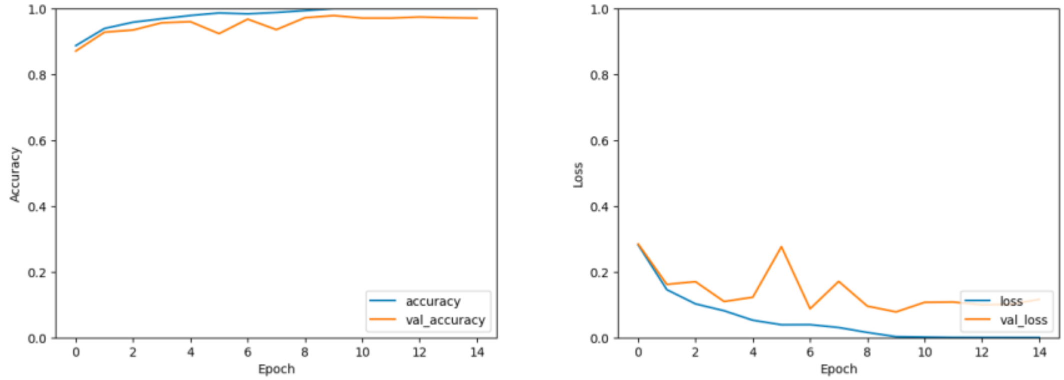
Model	Accuracy	Loss	Validation Accuracy	Validation Loss
Scratched CNN	0.9646	0.0968	0.9373	0.1875
MobileNetV2	1.0000	1.7341e-04	0.9714	0.1160
ResNet50	0.9295	0.1797	0.8790	0.2817
VGG19	0.9653	0.0998	0.9450	0.1453

Table 4.5 Accuracy Results for Unsharp Mask Filter.



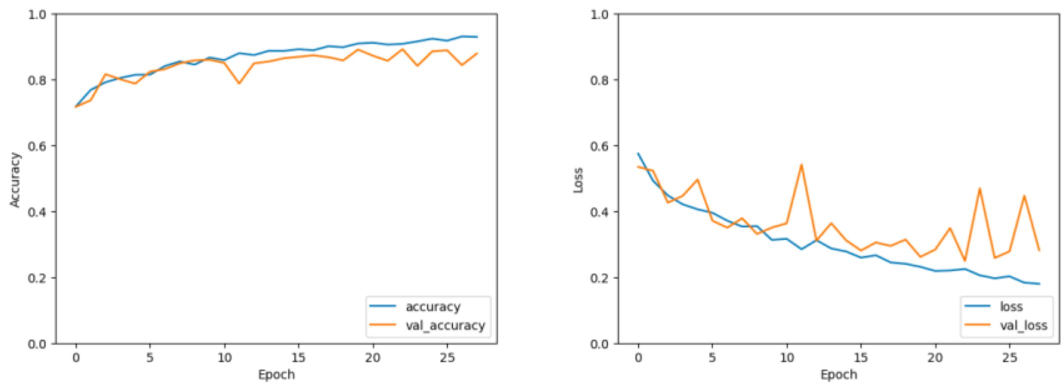
(a) Accuracy (b) Loss

Fig 4.7 Performance Results of Scratched CNN.



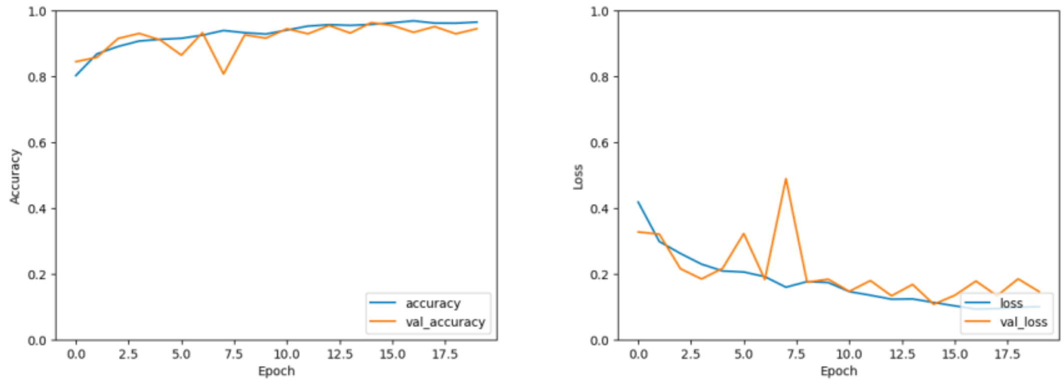
(a) Accuracy (b) Loss

Fig 4.8 Performance Results of MobileNetV2.



(a) Accuracy (b) Loss

Fig 4.9 Performance Results of ResNet50.



(a) Accuracy (b) Loss

Fig 4.10 Performance Results of VGG19.

4.2.2 Performance Metrics

Models	Class	Precision	Recall	F1-Score
Scratched CNN	Brain Tumor	98.67%	99.33%	99.01%
	Healthy	99.19%	98.40%	98.79%
MobileNetV2	Brain Tumor	87.11%	85.95%	86.53%
	Healthy	83.59%	84.92%	84.25%
ResNet50	Brain Tumor	94.01%	91.75%	92.87%
	Healthy	91.01%	93.46%	92.22%
VGG19	Brain Tumor	93.15%	90.97%	92.04%
	Healthy	89.57%	92.06%	90.80%

Table 4.6 Precision, Recall and F-1 Score for Random Forest Classifier

Models	Class	Precision	Recall	F1-Score
Scratched CNN	Brain Tumor	90.61%	99.33%	94.77%
	Healthy	99.09%	87.60%	92.99%
MobileNetV2	Brain Tumor	85.37%	95.65%	90.22%
	Healthy	93.98%	80.55%	86.75%
ResNet50	Brain Tumor	85.89%	92.09%	88.89%
	Healthy	90.37%	83.07%	86.57%
VGG19	Brain Tumor	89.81%	97.32%	93.41%
	Healthy	96.47%	86.90%	91.44%

Table 4.7 Precision, Recall and F-1 Score for KNN Classifier

Models	Class	Precision	Recall	F1-Score
Scratched CNN	Brain Tumor	97.31%	96.34%	96.83%
	Healthy	95.65%	96.80%	96.22%
MobileNetV2	Brain Tumor	78.35%	76.25%	77.28%
	Healthy	72.69%	75.00%	73.83%
ResNet50	Brain Tumor	84.75%	82.13%	83.42%
	Healthy	80.67%	83.46%	82.04%
VGG19	Brain Tumor	85.44%	89.79%	87.56%
	Healthy	87.60%	82.90%	84.97%

Table 4.8 Precision, recall and F-1 Score for AdaBoost Classifier

4.3 Analysis of CNN Models for Gaussian Filter

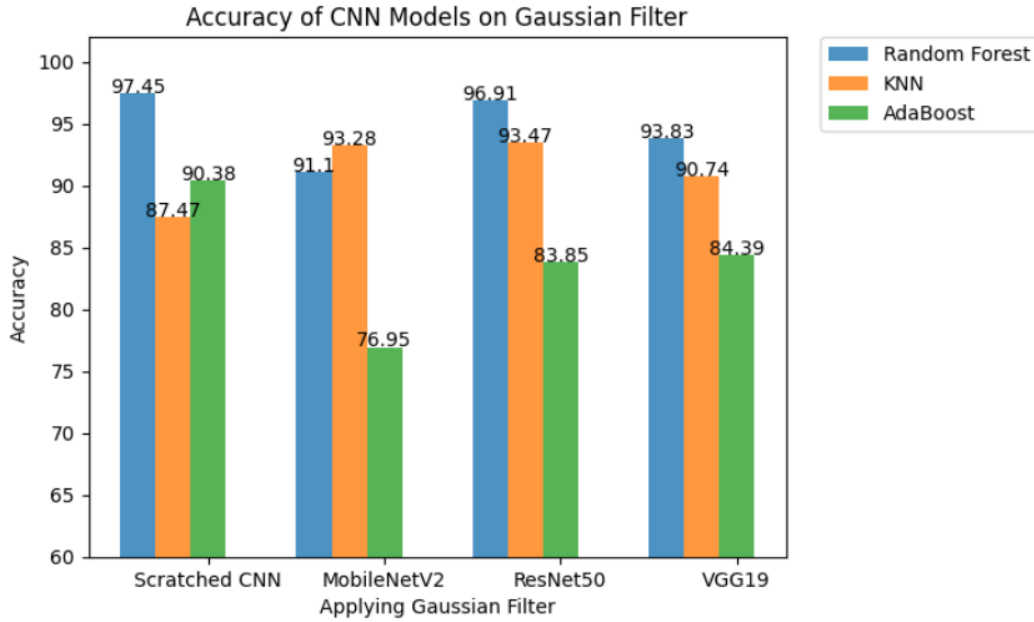


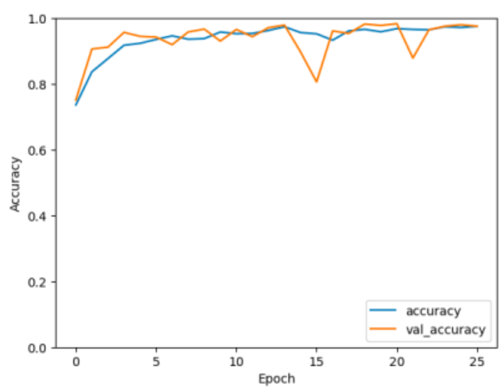
Fig 4.11 Histogram of the Accuracy of CNN Models on Gaussian Filter

The histogram Fig 4.11 delineates the accuracy of various CNN models on the Gaussian filter. The comparisons are carried out between Random Forest, KNN and AdaBoost classifiers in accordance with the model training implementing these classifiers. Scratched CNN using Random Forest has the greatest accuracy within all the models which is 97.45%. The accuracy of the Random Forest classifier is on the higher accuracy range compared to the other two classifiers having 92.47% and 93.83% when ResNet50 and VGG19 models were implemented. The accuracy of Random Forest is lowest in MobileNetV2 which is 91.10%. AdaBoost performs much worse than the other two classifiers in three of the models, MobileNetV2, ResNet50, and VGG19 with accuracy of 76.95%, 83.85%, and 84.39%, respectively. However, the accuracy of the AdaBoost classifier rose to 90.38% while using the Scratched CNN model. The KNN classifier is on the middle of the range in terms of accuracy for two of the models which are ResNet50 and VGG19 whereas it showed substantial growth when MobileNetV2 model has been used as it had the highest accuracy among all three classifiers which is 93.28%. Overall, it can be inferred that the use of different classifiers along with which model has been used to process a dataset have a compelling impact on the degrees of accuracy while using the same dataset.

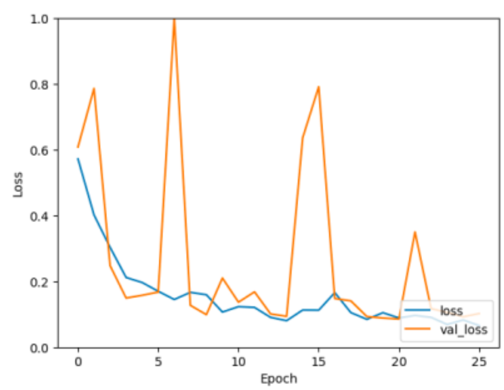
4.3.1 Performance Results

Model	Accuracy	Loss	Validation Accuracy	Validation Loss
Scratched CNN	0.9755	0.0655	0.9758	0.1021
MobileNetV2	0.9983	0.0050	0.9791	0.0711
ResNet50	0.7456	0.4978	0.7129	0.5168
VGG19	0.9499	0.1301	0.9560	0.1179

Table 4.9 Accuracy Results for Gaussian Filter.

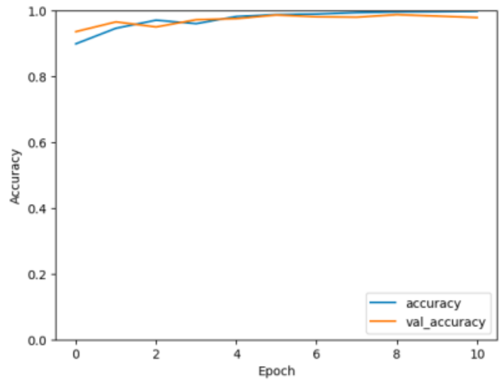


Accuracy

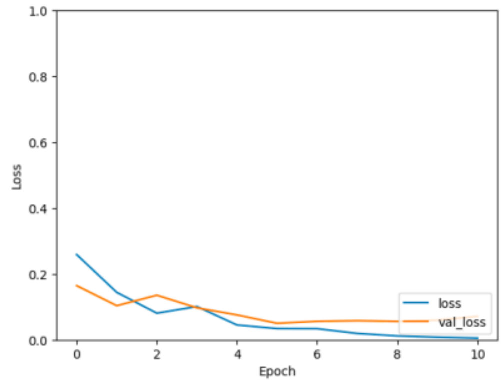


(b) Loss

Fig 4.12 Performance Results of Scratched CNN.

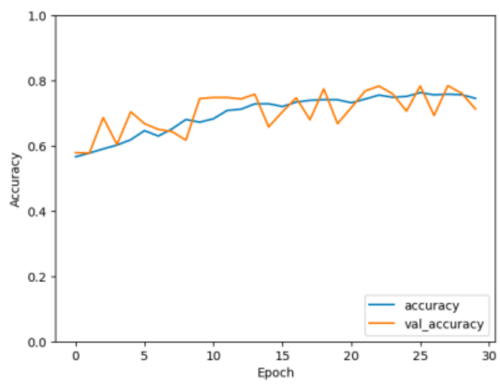


(a) Accuracy

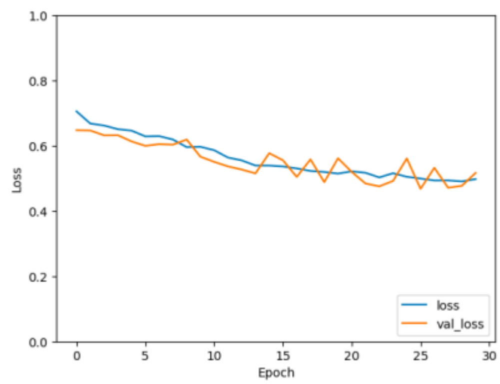


(b) Loss

Fig 4.13 Performance Results of MobileNetV2.

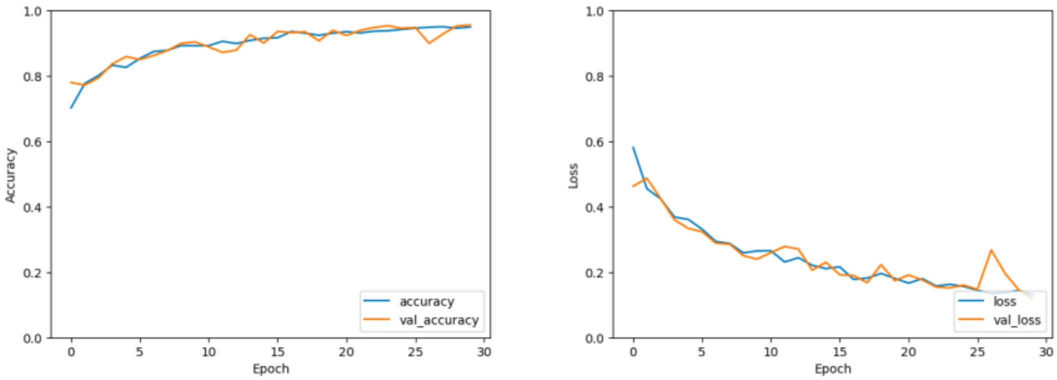


(a) Accuracy



(b) Loss

Fig 4.14 Performance Results of ResNet50.



(a) Accuracy (b) Loss

Fig 4.15 Performance Results of VGG19.

4.3.2 Performance Metrics

Models	Class	Precision	Recall	F1-Score
Scratched CNN	Brain Tumor	97.26%	97.93%	97.59%
	Healthy	97.68%	96.93%	97.30%
MobileNetV2	Brain Tumor	91.80%	91.49%	91.65%
	Healthy	90.31%	90.66%	90.46%
ResNet50	Brain Tumor	97.38%	97.06%	97.22%
	Healthy	96.32%	96.72%	96.52%
VGG19	Brain Tumor	95.39%	93.54%	94.46%
	Healthy	91.90%	94.19%	93.03%

Table 4.10 Precision, Recall and F-1 Score for Random Forest Classifier

Models	Class	Precision	Recall	F1-Score
Scratched CNN	Brain Tumor	85.53%	91.72%	88.51%
	Healthy	90.00%	82.76%	86.22%
MobileNetV2	Brain Tumor	92.69%	94.89%	93.78%
	Healthy	94.00%	91.44%	92.70%
ResNet50	Brain Tumor	91.95%	96.74%	94.28%
	Healthy	95.61%	89.34%	92.37%
VGG19	Brain Tumor	90.09%	93.87%	91.84%
	Healthy	91.67%	86.72%	89.12%

Table 4.11 Precision, Recall and F-1 Score for KNN Classifier

Models	Class	Precision	Recall	F1-Score
Scratched CNN	Brain Tumor	91.58%	90.00%	90.78%
	Healthy	89.09%	90.80%	89.94%
MobileNetV2	Brain Tumor	78.49%	78.23%	78.36%
	Healthy	75.19%	75.48%	75.34%
ResNet50	Brain Tumor	85.85%	85.01%	85.43%
	Healthy	81.37%	82.37%	81.87%
VGG19	Brain Tumor	89.16%	82.25%	85.57%
	Healthy	79.24%	87.13%	83.00%

Table 4.12 Precision, Recall and F-1 Score for AdaBoost Classifier

4.4 Analysis of CNN Models for Laplacian Filter

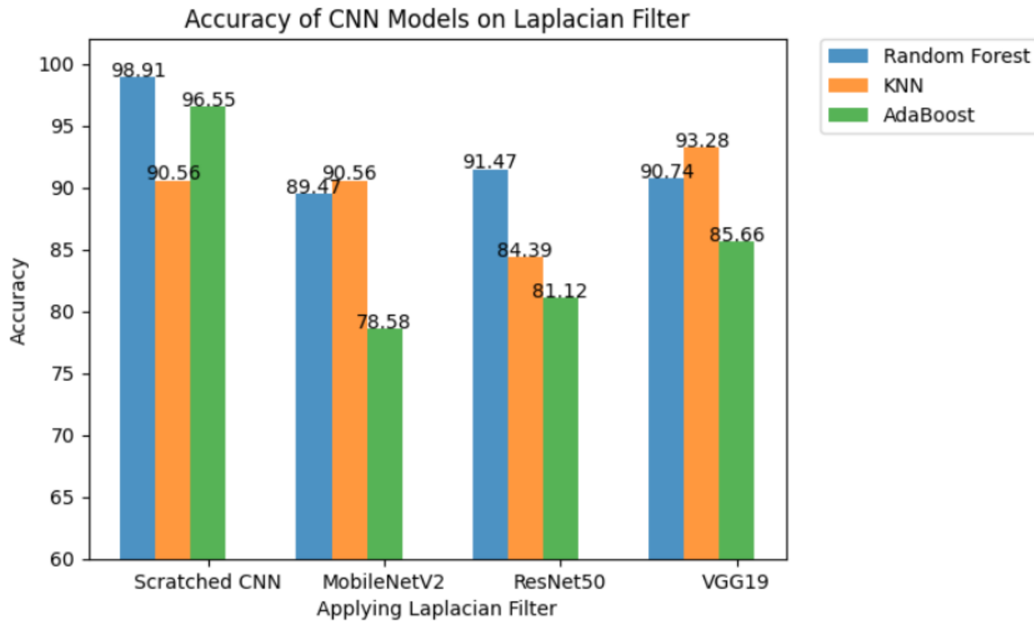


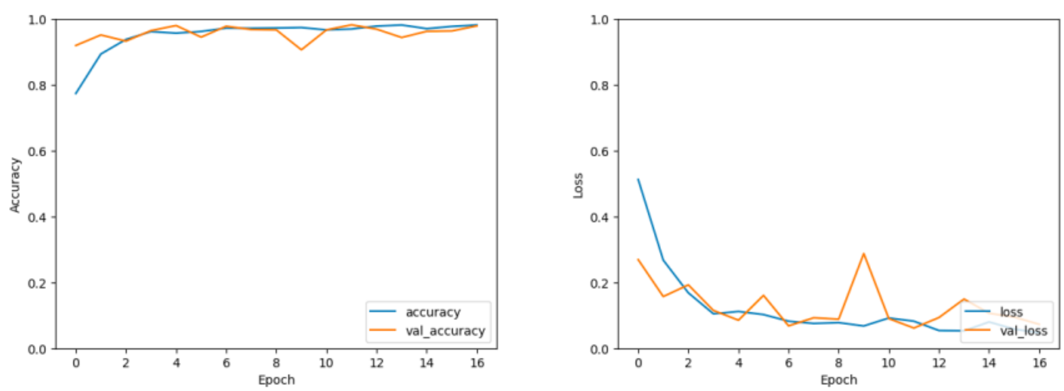
Fig 4.16 Histogram of the Accuracy of CNN Models on Laplacian Filter

The histogram Fig 4.16 delineates the accuracy of various CNN models on the Laplacian Filter. The comparisons are carried out between Random Forest, KNN and AdaBoost classifiers in accordance with the model training implementing these classifiers. Scratched CNN using Random Forest has the greatest accuracy within all the models which is 98.91%. The accuracy of the Random Forest classifier is on the higher accuracy range compared to the other two classifiers having 91.47% and 90.74% when ResNet50 and VGG19 models were implemented. The accuracy of Random Forest is lowest in MobileNetV2 which is 89.47%. AdaBoost performs much worse than the other two classifiers in three of the models, MobileNetV2, ResNet50, and VGG19 with accuracy of 78.58%, 81.12%, and 85.66%, respectively. However, the accuracy of the AdaBoost classifier rose to 96.55% while using the Scratched CNN model. The KNN classifier is on the middle of the range in terms of accuracy for two of the models which are ResNet50 and VGG19 whereas it showed substantial growth when MobileNetV2 model has been used as it had the highest accuracy among all three classifiers which is 90.56%. Overall, it can be inferred that the use of different classifiers along with which model has been used to process a dataset have a compelling impact on the degrees of accuracy while using the same dataset.

4.4.1 Performance Results:

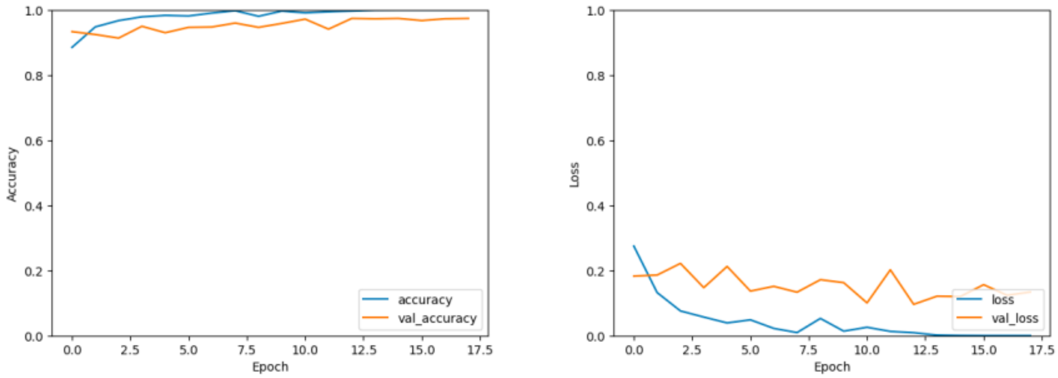
Model	Accuracy	Loss	Validation Accuracy	Validation Loss
Scratched CNN	0.9819	0.0469	0.9791	0.0729
MobileNetV2	1.0000	9.8622e-05	0.9747	0.1339
ResNet50	0.8692	0.3061	0.8020	0.4515
VGG19	0.9496	0.1349	0.9087	0.2225

Table 4.13 Accuracy Results for Laplacian Filter.



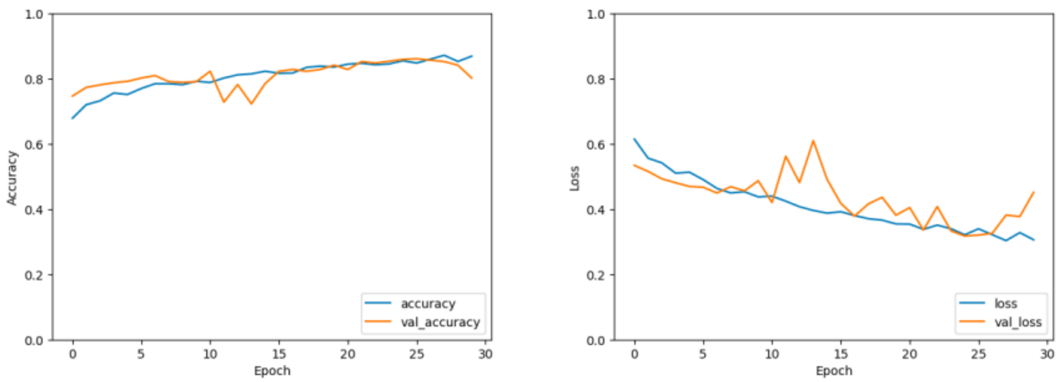
a. Accuracy (b) Loss

Fig 4.17 Performance Results of Scratched CNN.



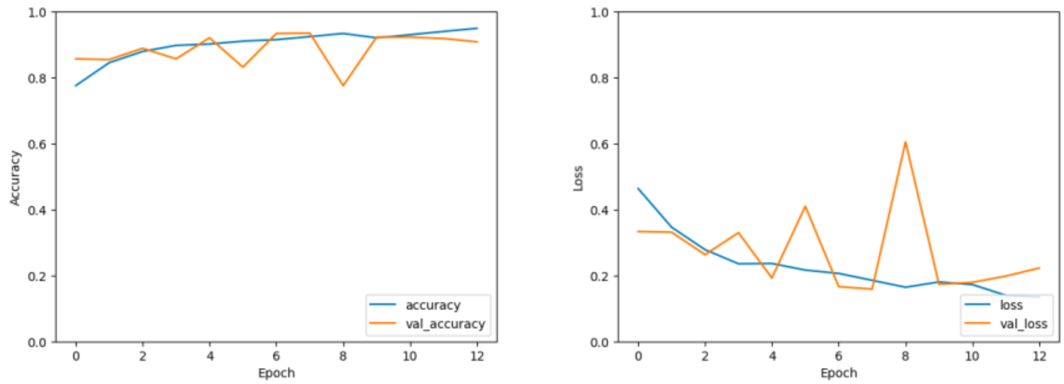
(a) Accuracy (b) Loss

Fig 4.18 Performance Results of MobileNetV2.



(a) Accuracy (b) Loss

Fig 4.19 Performance Results of ResNet50.



(a) Accuracy (b) Loss

Fig 4.20 Performance Results of VGG19.

4.4.2 Performance Metrics

Models	Class	Precision	Recall	F1-Score
Scratched CNN	Brain Tumor	99.65%	98.27%	98.95%
	Healthy	98.11%	99.61%	98.85%
MobileNetV2	Brain Tumor	92.10%	89.17%	90.61%
	Healthy	86.23%	89.87%	88.01%
ResNet50	Brain Tumor	91.57%	94.11%	92.82%
	Healthy	91.32%	87.71%	89.48%
VGG19	Brain Tumor	91.00%	91.91%	91.45%
	Healthy	90.44%	89.37%	89.90%

Table 4.14 Precision, Recall and F-1 Score for Random Forest Classifier

Models	Class	Precision	Recall	F1-Score
Scratched CNN	Brain Tumor	87.18%	96.20%	91.47%
	Healthy	95.24%	84.29%	89.43%
MobileNetV2	Brain Tumor	87.86%	96.81%	92.21%
	Healthy	95.12%	82.27%	88.23%
ResNet50	Brain Tumor	82.46%	93.18%	87.50%
	Healthy	88.17%	71.92%	79.22%
VGG19	Brain Tumor	90.88%	97.30%	93.98%
	Healthy	96.56%	88.85%	92.40%

Table 4.15 Precision, Recall and F-1 Score for KNN Classifier

Models	Class	Precision	Recall	F1-Score
Scratched CNN	Brain Tumor	97.21%	96.20%	96.71%
	Healthy	95.83%	96.93%	96.38%
MobileNetV2	Brain Tumor	83.10%	78.34%	80.65%
	Healthy	73.33%	78.90%	76.01%
ResNet50	Brain Tumor	83.69%	84.21%	83.95%
	Healthy	77.43%	76.75%	77.09%
VGG19	Brain Tumor	86.09%	87.54%	86.81%
	Healthy	85.14%	83.46%	84.29%

Table 4.16 Precision, Recall and F-1 Score for AdaBoost Classifier

4.5 Result Analysis

As discussed, we used three filters in this research: Unsharp Mask, Gaussian and Laplacian filter. Pre-trained models like VGG16, ResNet50, MobileNetV2 and Scratched CNN have been used for feature extraction. For classification, we used Random Forest, KNN and AdaBoost classifiers. Fig 4.1 represents the original dataset's accuracy without using any filter with the CNN models. Among all the models, Scratch CNN with Random Forest classifier gives the best Accuracy, 97.45%. From Fig 4.6, it is observed that in the dataset Unsharp Mask filter has been used, and here, Scratch CNN with Random Forest achieved the highest accuracy among other CNN models. The accuracy is 98.81%. In Fig 4.11, it can be seen that the Gaussian filter has been used in the dataset, and the best accuracy is 97.45%. Moreover, the Scratch CNN model with Random Forest gains the highest accuracy again. The last figure, Fig 4.16, portrays that the Laplacian filter has been applied in the dataset. The highest accuracy is 98.91%, again achieved by the Scratch CNN model with the Random Forest classifier.

The figures 4.1, 4.6, 4.11 and 4.16 show that the Laplacian filter with Scratch CNN and Random Forest classifiers achieved the best accuracy among all the filters, models and classifiers. The highest accuracy is 97.81%. Table 4.17 is given below for a better understanding.

Dataset	Best CNN Model	Best Classifier	Accuracy
Original Dataset	Scratched CNN	Random Forest	97.45%
Unsharp Mask	Scratched CNN	Random Forest	98.81%
Laplacian	**Scratched CNN**	**Random Forest**	**98.91%**
Gaussian	Scratched CNN	Random Forest	97.45%

Table 4.17 Best Model and Classifier for Every Dataset

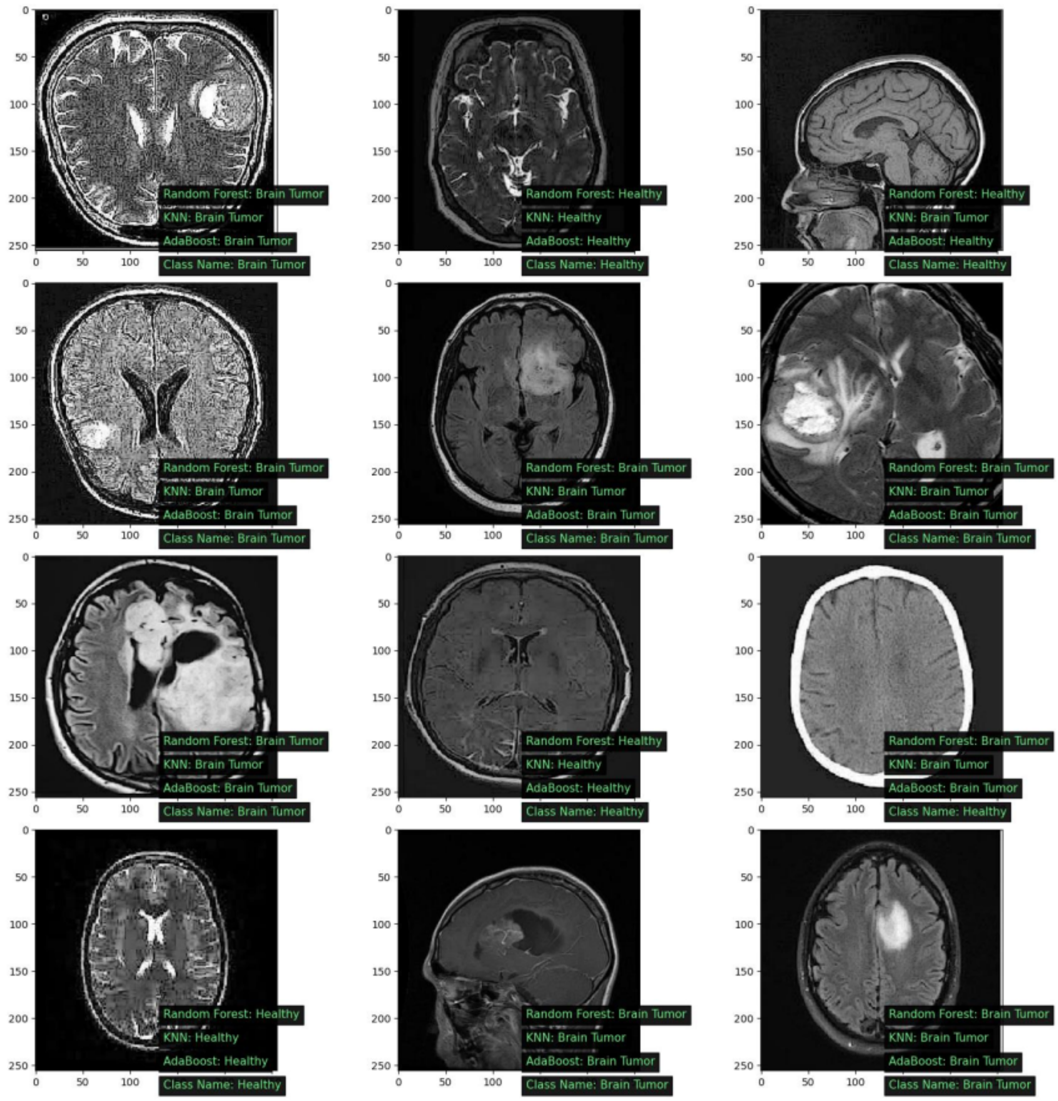


Fig 4.21 Visual Representation of Prediction

Figure 4.21 represents model validation by random test images in the Scratched cnn model for the Laplacian filter, showing the result for every classifier.

4.6 Impact of Filters

4.6.1 Modified Scratched CNN

The Scratched CNN model achieved an original accuracy of 97.45%. When the Unsharp Mask Filter was applied to the data, the accuracy improved slightly to 98.81%. However, the Gaussian Filter did not have a noticeable effect, maintaining the same accuracy as the original. On the other hand, the Laplacian Filter resulted in a slight improvement, with an accuracy of 98.91%. These findings suggest that the Laplacian Filter could be a useful enhancement for the Scratched CNN model.

4.6.2 Modified MobileNetV2

The MobileNetV2 model had an original accuracy of 87.84%. When the Unsharp Mask Filter was applied, the accuracy decreased to 85.48%. This indicates that the filter may not be suitable for improving the performance of this particular model. However, the Gaussian Filter showed promise by increasing the accuracy to 91.10%. The Laplacian Filter, on the other hand, resulted in a slight decrease in accuracy to 89.47%. These results highlight the potential of the Gaussian Filter for enhancing the accuracy of the MobileNetV2 model.

4.6.3 Modified ResNet50

The ResNet50 model achieved an original accuracy of 92.37%. Applying the Unsharp Mask Filter resulted in a slight improvement to 92.55%, indicating that this filter had a limited impact on the model's performance. However, the Gaussian Filter showed a more significant increase in accuracy to 96.91%. The Laplacian Filter, while still maintaining a relatively high accuracy of 91.17%, resulted in a slight decrease compared to the original accuracy. These findings suggest that the Gaussian Filter is the most effective in enhancing the ResNet50 model's accuracy among the filters considered.

4.6.4 Modified VGG19

The VGG19 model had an original accuracy of 94.74%. When the Unsharp Mask Filter was applied, the accuracy decreased to 91.47%, indicating that this filter had a negative impact on the model's performance. The Gaussian Filter, while showing a minor improvement to 93.83%, did not significantly enhance the accuracy compared to the original. The Laplacian Filter, however, resulted in a noticeable decrease in accuracy to 90.74%. These findings suggest that the filters applied did not effectively enhance the performance of the VGG19 model in this particular scenario.

Overall, the impact of different filters on the accuracy of the models varied. The Unsharp Mask Filter showed mixed results across the models, with both improvements and decreases in accuracy. The Gaussian Filter demonstrated the most promising results by improving accuracy for certain models, particularly ResNet50. The Laplacian Filter had a mixed impact, with both slight improvements and decreases in accuracy observed. However, it is important to note that these findings are specific to the provided dataset and models and may not be generalizable to other datasets or models.

Chapter 5

Discussion

This study proposes a novel approach to detecting brain tumors using image processing techniques with MRI images. Filtering removes the image's noise and clearly identifies the tumor part of the brain. Classifiers categorize and label the images and, in this thesis, Random Forest, KNN and AdaBoost classifiers were used. For filtering, Gaussian, Unsharp mask and Laplacian filters were used. Scratch CNN, ResNet50 and MobileNetV2 models were used for feature extraction. Scratched CNN and Random Forest perform better than other models and classifiers in all four datasets. In this study, the highest accuracy was achieved by the Scratch CNN model with Random Forest classifier on the Laplacian dataset. The accuracy is 98.91%. The second-best dataset is Unsharp Mask, where Scratch CNN and Random Forest gained 98.81% accuracy. This study also analyzes different CNN models' F1 score, Recall and Precision.

5.1 Limitations

Considering the improvements and encouraging outcomes described in this work, there are a number of limitations that need to be acknowledged:

5.1.1 Limited Dataset Size

The study may have used a relatively small dataset, which could have an impact on how generalizable the findings are. Access to larger and more varied datasets could help to further test the suggested method and offer a more thorough examination.

5.1.2 Scope of Tumor Types

The study may concentrate on a certain subset of brain tumor types, which would restrict its applicability to a wider variety of tumor kinds. The usefulness of the suggested strategy would be improved by broadening the scope to incorporate different tumor types.

5.1.3 Limited Clinical Validation

Although the suggested method is highly accurate at finding brain tumors, it may not have received sufficient clinical confirmation. To evaluate how well it performs in real-world circumstances, thorough clinical trials and validation studies involving medical experts and actual patient data would be required.

5.1.4 Limited Comparison with Existing Methods

Despite the proposed method's notable accuracy, there may not be a direct comparison to other cutting-edge approaches now in use. Comparative research with other proven techniques would offer a standard against which to measure the effectiveness and efficiency of the system under consideration.

5.1.5 Hardware and Computational Requirements

The computational and hardware requirements necessary to implement the suggested approach may not be covered in the study. The system's deployment and real-time implementation may be hampered by resource or computational constraints.

Chapter 6

Conclusion

Image processing plays a significant role in analyzing various diseases in the sector of modern medical science. This study involves the testing and identification of brain tumors using MRI images. Brain tumors are detected and classified with machine learning classifiers and image processing. The scratch Convolutional neural network (CNN) model and Random Forest classifier outperform the other techniques in terms of accuracy in identifying brain tumors. This combination gave the best results in our research.

6.1 Future Work

In the future, we will aim to enhance our accuracy in detecting brain tumors by implementing more advanced methods and techniques. We will work on specific tumor datasets like Meningioma, Pituitary Adenoma, Craniopharyngioma, etc. We plan to work on other medical images like CT scans and Xray. We hope that this work will be beneficial for the doctors and physicians to make decisions for their final treatment. This work will help to implement a predictive accuracy model into Magnetic Resonance Imaging (MRI) systems so that when patients receive their MRI results, they will also be provided with information on the accuracy of the predictions made by the model. From this study, researchers can have an idea about the best filters for brain tumor MRI images. It is anticipated that by pursuing these future research directions, the brain tumor detection system's accuracy, effectiveness, and clinical applicability can be further improved, leading to better patient outcomes and assisting medical professionals in their diagnostic decision-making process.

Bibliography

[1] Mrs K. Padmavathi, Ms C.Megala, ”Detection of Brain Tumour with Filtering Techniques“, July 2015

[2] Brain Tumor: Statistics, Cancer.Net Editorial Board (Accessed on 5th May 2023) https://www.cancer.net/cancer-types/brain-tumor/statistics

[3] Nan Wu; Jason Phang; Jungkyu Park; Yiqiu Shen; Zhe Huang; Masha Zorin; Stanisław Jastrzębski. Deep Neural Networks Improve Radiologists’ Performance in Breast Cancer Screening. 07 October 2019

[4] National Brain Tumor Society (NBTS). Editorial Board (Accessed on 5th May 2023) , https://braintumor.org/brain-tumors/about-brain-tumors/brain-tumor-facts/

[5] Praveen Gamage, “Identification of Brain Tumor using Image Processing Techniques - Image Pre-Processing and Segmentation”, 2017.

[6] Dunsheng Liu, Yuanning Liu, Liyan Dong, “ResNet: Improved ResNet for Brain Tumor Classification “, December 2019

[7] Ouiza Nait Belaid, Malik Loudini, “Classification of Brain Tumor by Combination of Pre-Trained VGG16 CNN”, 2020

[8] Ramya Mohan, K. Ganapathy, R. A, “Brain tumor classification of magnetic resonance images using a novel CNN-based medical image analysis and detection network in comparison to VGG16”. (2022)

[9] Zahid Rasheed, Yong-Kui Ma, Inam Ullah, Tamara Al Shloul, Ahsan Bin Tufail, Yazeed Yasin Ghadi, Muhammad Zubair Khan and Heba G. Mohamed, “Automated Classification of Brain Tumors from Magnetic Resonance Imaging Using Deep Learning”, 2023

[10] Arpit Kumar Sharma, Amita Nandal, Arvind Dhaka, Kemal Polat, Raghad Alwadie, Fayadh Alenezi, Adi Alhudhaif, Adi Alhudhaif, “HOG transformation-based feature extraction framework in modified Resnet50 model for brain tumor detection”, July 2023

[11] Tonmoy Hossain1, Fairuz Shadmani Shishir, Mohsena Ashraf, MD Abdullah Al Nasim, Faisal Muhammad Shah, “Brain Tumor Detection Using Convolutional Neural Network”, 2019

[12] Aryan Sagar Methil, Brain Tumor Detection using Deep Learning and Image Processing. March 2021

[13] A. Ari, D. Hanbay, Deep learning based brain tumor classification and detection system. 2018

[14] Justin S. Paul, A. Plassard, B. Landman, D. Fabbri, Deep learning for brain tumor classification, March 2017

[15] Amjad Rehman, Muhammad Attique Khan, Tanzila Saba, Zahid Mehmood, Usman Tariq, Noor Ayesha, Microscopic brain tumor detection and classification using 3D CNN and feature selection architecture. September 2020

[16] Liya Zhao, Ke-bin Jia, Multiscale CNNs for Brain Tumor Segmentation and Diagnosis. 2016

[17] Anand Deshpande, Vania V. Estrela, Prashant Patavardhan, The DCT-CNN-ResNet50 architecture to classify brain tumours with super-resolution, convolutional neural network, and the ResNet50. December 2021

[18] Imran Javaid, Shuai Zhang, A. E. K. Isselmou, Souha Kamhi, Isah Salim Ahmad, Brain Tumor Classification & Segmentation by Using Advanced DNN, CNN & ResNet-50 Neural Networks. 2020

[19] H. Khan, Wu Jue, Muhammad Mushtaq, Muhammad Umer Mushtaq, Brain tumor classification in MRI image using convolutional neural network. 2020

[20] Ahmet Çinar, Muhammed Yildirim, Detection of tumors on brain MRI images using the hybrid convolutional neural network architecture. 2020

[21] Zhiguan Huang, Xiaohao Du, Liangming Chen, Yuhe Li, Mei Liu, Yao Chou, Convolutional Neural Network Based on Complex Networks for Brain Tumor Image Classification With a Modified Activation Function. 2020

[22] Yakub Bhanothu, Anandhanarayanan Kamalakannan, G. Rajamanickam, Detection and Classification of Brain Tumor in MRI Images using Deep Convolutional Network. 2020

[23] A. P. Rahmathunneesa, K. V. Ahammed Muneer, Performance Analysis of Pre-trained Deep Learning Networks for Brain Tumor Categorization. 2019

[24] Tahia Tazin, S. Sarker, P. Gupta, Fozayel Ibn Ayaz, Sumaia Islam, A Robust and Novel Approach for Brain Tumor Classification Using Convolutional Neural Network

[25] Divjot Kaur, Silky Goel, Rahul Nijhawan, Siddharth Gupta, Analysis of Brain Tumor using Pre-trained CNN Models and Machine Learning Techniques. 2022

[26] Roshan Kumari, Saurabh Kumar Srivastava Machine Learning: A Review on Binary Classification. February 2017

[27] Javaria Amin, Muhammad Sharif, Anandakumar Haldorai, Mussarat Yasmin & Ramesh Sundar Nayak Brain tumor detection and classification using machine learning: a comprehensive survey. 08 November 2021

[28] Muhammad Usman Saeed, Ali Ghulam, Wang Bin, Sultan Almotiri. RMU-Net: A Novel Residual Mobile U-Net Model for Brain Tumor Segmentation from MR Images. August 2021

[29] Venkatesan Rajinikanth, Alex Noel Joseph Raj, Ganesh Naik, Palani Thanaraj Krishnan. A Customized VGG19 Network with Concatenation of Deep and Handcrafted Features for Brain Tumor Detection. May 2020

[30] Lichi Zhang, Han Zhang, Islem Rekik, Yaozong Gao, Qian Wang & Dinggang Shen. Malignant Brain Tumor Classification Using the Random Forest Method. 02 August 2018

[31] Bushra Naz, Nuzhat Faraz, Data Mining Approach for Detection and Classification of Brain Tumor. January 2022

[32] Bijen Khagi, Goo Rak Kwon, "Convolutional Neural Network-Based Natural Image and MRI Classification Using Gaussian Activated Parametric (GAP) Layer", Access IEEE, vol. 9, pp. 96930-96947, 2021

[33] Rajinikanth V, Satapathy SC, Fernandes SL, Nachiappan S. Entropy based segmentation of tumor from

brain MR images – a study with teaching learning based optimization. Pattern Recogn Lett 2017;94:87–95. Google Scholar

[34] Ramya Mohan, Kirupa Ganapathy, Rama A . Brain tumour classification of magnetic resonance images using a novel CNN-based medical image analysis and detection network in comparison to VGG16. January 2022

[35] Muhammad Asad Arshed, Aqsa Shahzad, Kamran Arshad Multiclass Brain Tumor Classification from MRI Images using Pre-Trained CNN Model. November 2022

[36] P. S. Lathashree, Prajna Puthran, B. N. Yashaswini & Nagamma Patil. Brain MR Image Segmentation for Tumor Identification Using Hybrid of FCM Clustering and ResNet.2 august 2022

[37] Muhammad Usman Saeed, Ali Ghulam, Wang Bin, Sultan Almotiri. RMU-Net: A Novel Residual Mobile U-Net Model for Brain Tumor Segmentation from MR Images. August 2021

[38] Muhammad Asad Arshed, Aqsa Shahzad, Kamran Arshad Multiclass Brain Tumor Classification from MRI Images using Pre-Trained CNN Model. November 2022

[39] Venkatesan Rajinikanth, Alex Noel Joseph Raj, Ganesh Naik, Palani Thanaraj Krishnan. A Customized VGG19 Network with Concatenation of Deep and Handcrafted Features for Brain Tumor Detection. May 2020

[40] Gayathri Shrikanth & Sanika Mhadgut. Brain Tumor Detection using FastAI and OpenCV. May 5, 2020. Brain Tumor Detection using FastAI and OpenCV

[41] Anindya Apriliyanti Pravitasari, Nur Iriawan UNet-VGG16 with transfer learning for MRI-based brain tumor segmentation

[42] Vania V. Estrela, Anand Deshpande, Prashant Patavardhan. The DCT-CNN-ResNet50 architecture to classify brain tumors with super-resolution, convolutional neural network, and the ResNet50 . October 2021

[43] Martin Abadi Paul Barham Jianmin Chen Zhifeng Chen Andy Davis Jeffrey Dean Matthieu Devin Sanjay Ghemawat Geoffrey Irving Michael Isard Manjunath Kudlur Josh Levenberg Rajat Monga Sherry Moore Derek G. Murray Benoit Steiner Paul Tucker Vijay Vasudevan Pete Warden Martin Wicke Yuan Yu Xiaoqiang Zheng.TensorFlow: A system for large-scale machine learning.2016.

[44] Fabian Pedregosa; Gael Varoquaux, Vincent Michael. Scikit-learn: Machine Learning in Python. January 2012

[45] Ivan Culjak; David Abram; Tomislav Pribanic; Hrvoje Dzapo; Mario Cifrek. A brief introduction to OpenCV. 16 July 2012

Brain Tumor Detection Using Deep Learning Techniques

ORIGINALITY REPORT

19%
SIMILARITY INDEX

PRIMARY SOURCES

1	www.researchgate.net Internet	276 words — 2%
2	mountainscholar.org Internet	138 words — 1%
3	worldwidescience.org Internet	124 words — 1%
4	Jia Chen, Shihui Duan, Fei Long, Yongxing Wang, Song Wang, Qiang Ling. "Two-Stream Convolutional Neural Networks for Emergency Recognition in Images", 2020 39th Chinese Control Conference (CCC), 2020 Crossref	98 words — 1%
5	www.mdpi.com Internet	82 words — 1%
6	www.journaltocs.ac.uk Internet	61 words — < 1%
7	"Artificial Intelligence and Smart Environment", Springer Science and Business Media LLC, 2023 Crossref	54 words — < 1%
8	www.semanticscholar.org Internet	51 words — < 1%

9	"Proceedings of International Conference on Recent Trends in Computing", Springer Science and Business Media LLC, 2022 Crossref	50 words — < 1%
10	webthesis.biblio.polito.it Internet	45 words — < 1%
11	deeplizard.com Internet	44 words — < 1%
12	www.coursehero.com Internet	44 words — < 1%
13	Nahid Ferdous Aurna, Mohammad Abu Yousuf, Kazi Abu Taher, A.K.M. Azad, Mohammad Ali Moni. "A classification of MRI brain tumor based on two stage feature level ensemble of deep CNN models", Computers in Biology and Medicine, 2022 Crossref	43 words — < 1%
14	repositorio.comillas.edu Internet	39 words — < 1%
15	Hari Kishan Kondaveeti, Kalyan Gandhi Ujini, Bikkina Veera Venkata Pavankumar, Bollu Sai Tarun, Simhadri Chinna Gopi. "Plant Disease Detection Using Ensemble Learning", 2023 2nd International Conference on Computational Systems and Communication (ICCSC), 2023 Crossref	36 words — < 1%
16	ntnuopen.ntnu.no Internet	36 words — < 1%
17	utpedia.utp.edu.my Internet	36 words — < 1%

18	lib.buet.ac.bd:8080 Internet	32 words — < 1%
19	Mohamed Hammad, Mohammed ElAffendi, Abdelhamied A. Ateya, Ahmed A. Abd El-Latif. "Efficient Brain Tumor Detection with Lightweight End-to-End Deep Learning Model", Cancers, 2023 Crossref	31 words — < 1%
20	João Pedro Fontes, João Nuno Centeno Raimundo, Luís Gonzaga Mendes Magalhães, Miguel Angel Guevara Lopez. "3D CNN-based Quantitative Imaging Biomarkers for Accurate Phenotyping of Luminal A Breast Cancer in Magnetic Resonance Imaging", Institute of Electrical and Electronics Engineers (IEEE), 2023 Crossref Posted Content	28 words — < 1%
21	S.J.A. Jairam, D. Lokeshwar, B. Divya, P. Mohamed Fathimal. "Brain Tumor Detection Using Deep Learning", Trans Tech Publications, Ltd., 2023 Crossref	28 words — < 1%
22	Arpit Kumar Sharma, Amita Nandal, Arvind Dhaka, Kemal Polat, Raghad Alwadie, Fayadh Alenezi, Adi Alhudhaif. "HOG transformation based feature extraction framework in modified Resnet50 model for brain tumor detection", Biomedical Signal Processing and Control, 2023 Crossref	27 words — < 1%
23	A P Rahmathunneesa, K V Ahammed Muneer. "Performance Analysis of Pre-trained Deep Learning Networks for Brain Tumor Categorization", 2019 9th International Conference on Advances in Computing and Communication (ICACC), 2019 Crossref	26 words — < 1%

24 Alpamis Kutlimuratov, Jamshid Khamzaev, Temur Kuchkorov, Muhammad Shahid Anwar, Ahyoung Choi. "Applying Enhanced Real-Time Monitoring and Counting Method for Effective Traffic Management in Tashkent", Sensors, 2023 — 25 words — < 1%
Crossref

25 jitm.ut.ac.ir — 25 words — < 1%
Internet

26 "Communication Software and Networks", Springer Science and Business Media LLC, 2021 — 22 words — < 1%
Crossref

27 www.teledyneanalytical.com — 22 words — < 1%
Internet

28 Nematzadeh, Peyman. "Development of In-Field Data Acquisition Systems and Machine Learning-Based Data Processing and Analysis Approaches for Turfgrass Quality Rating and Peanut Flower Detection", Oklahoma State University, 2023 — 20 words — < 1%
ProQuest

29 pymed.ai — 19 words — < 1%
Internet

30 sciencepubco.com — 19 words — < 1%
Internet

31 www.knowledgehut.com — 19 words — < 1%
Internet

32 Asta Margienė. "Adaptyviųjų ir kompetencijomis grįstų žinių vertinimo sistemų duomenų struktūrų modeliai ir jų susiejimas", Vilnius Gediminas Technical University, 2023 — 18 words — < 1%

Crossref Posted Content

33 Pawan Kumar Mall, Pradeep Kumar Singh. "MTR-SDL: a soft computing based multi-tier rank model for shoulder X-ray classification", Soft Computing, 2023 — 18 words — < 1%
Crossref

34 Tok, Teck Bok. "Removing unimportant computations in interprocedural program analysis", Proquest, 20111109 — 18 words — < 1%
ProQuest

35 bura.brunel.ac.uk — 18 words — < 1%
Internet

36 www.botreetechnologies.com — 18 words — < 1%
Internet

37 "Inventive Systems and Control", Springer Science and Business Media LLC, 2022 — 17 words — < 1%
Crossref

38 Md. Nahiduzzaman, Md. Omaer Faruq Goni, Md. Shamim Anower, Md. Robiul Islam et al. "A Novel Method for Multivariant Pneumonia Classification based on Hybrid CNN-PCA Based Feature Extraction using Extreme Learning Machine with Chest X-Ray Images", IEEE Access, 2021 — 17 words — < 1%
Crossref

39 Rushil Anirudh, Vinay Venkataraman, Karthikeyan Natesan Ramamurthy, Pavan Turaga. "A Riemannian Framework for Statistical Analysis of Topological Persistence Diagrams", 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2016 — 17 words — < 1%
Crossref

40 www.carcinoid.org — 17 words — < 1%
Internet

41 "Understanding COVID-19: The Role of Computational Intelligence", Springer Science and Business Media LLC, 2022
Crossref
16 words — < 1%

42 Asma Naseer, Tahreem Yasir, Arifah Azhar, Tanzeela Shakeel, Kashif Zafar. "Computer-Aided Brain Tumor Diagnosis: Performance Evaluation of Deep Learner CNN Using Augmented Brain MRI", International Journal of Biomedical Imaging, 2021
Crossref
16 words — < 1%

43 www.ieindia.org
Internet
16 words — < 1%

44 "Advances in Computational Intelligence", Springer Science and Business Media LLC, 2019
Crossref
15 words — < 1%

45 Md Mushfiqur Rahman, Pratik Barua. "A CNN Model-based Ensemble Approach for Fruit Identification Using Seed", 2021 5th International Conference on Electrical Information and Communication Technology (EICT), 2021
Crossref
15 words — < 1%

46 jptcp.com
Internet
15 words — < 1%

47 towardsdatascience.com
Internet
15 words — < 1%

48 "Neural Information Processing", Springer Science and Business Media LLC, 2017
Crossref
14 words — < 1%

49	Momina Masood, Rabbia Maham, Ali Javed, Usman Tariq, Muhammad Attique Khan, Seifedine Kadry. "Brain MRI analysis using deep neural network for medical of internet things applications", Computers and Electrical Engineering, 2022 Crossref	14 words — < 1%
50	export.arxiv.org Internet	14 words — < 1%
51	github.com Internet	14 words — < 1%
52	www.simplilearn.com Internet	14 words — < 1%
53	"Smart Intelligent Computing and Communication Technology", IOS Press, 2021 Crossref	13 words — < 1%
54	Hu, Weihua. "On the Predictive Power of Graph Neural Networks", Stanford University, 2023 ProQuest	13 words — < 1%
55	www.cqc.org.uk Internet	13 words — < 1%
56	"Advanced Machine Learning Approaches in Cancer Prognosis", Springer Science and Business Media LLC, 2021 Crossref	12 words — < 1%
57	Anand, Deepa, and Sushma Ravichandran. "Investigations into the Goodness of Posts in Q&A Forums—Popularity Versus Quality", Advances in Intelligent Systems and Computing, 2015. Crossref	12 words — < 1%

58 Hayder Q. Flayyih, Jumana Waleed, Saad Albawi. "ASystematic Mapping Study on Brain Tumors Recognition Based on Machine Learning Algorithms", 2020 3rd International Conference on Engineering Technology and its Applications (IICETA), 2020 — 12 words — < 1%
Crossref

59 Tapesh Kumar, Puneet Kumar Yadav, Vrinda Yadav. "Detection of Brain Tumor using CNN", 2022 4th International Conference on Inventive Research in Computing Applications (ICIRCA), 2022 — 12 words — < 1%
Crossref

60 Zhiguan Huang, Xiaohao Du, Liangming Chen, Yuhe Li, Mei Liu, Yao Chou, Long Jin. "Convolutional Neural Network Based on Complex Networks for Brain Tumor Image Classification With a Modified Activation Function", IEEE Access, 2020 — 12 words — < 1%
Crossref

61 digitalcommons.wayne.edu — 12 words — < 1%
Internet

62 ijircce.com — 12 words — < 1%
Internet

63 saurabhg.com — 12 words — < 1%
Internet

64 Katukuri Arun Kumar, Ravi Boda. "A computer-aided brain tumor diagnosis by adaptive fuzzy active contour fusion model and deep fuzzy classifier", Multimedia Tools and Applications, 2022 — 11 words — < 1%
Crossref

65 globbac.firebaseapp.com — 11 words — < 1%
Internet

66	www.dailyexcelsior.com Internet	11 words — < 1%
67	www.medrxiv.org Internet	11 words — < 1%
68	Narayan, Ashwin. "Real-Time Hierarchical Classification of Motion Intent for Wearable Robotics", National University of Singapore (Singapore), 2023 ProQuest	10 words — < 1%
69	assets.researchsquare.com Internet	10 words — < 1%
70	elib.bsu.by Internet	10 words — < 1%
71	krex.k-state.edu Internet	10 words — < 1%
72	labs.cd2h.org Internet	10 words — < 1%
73	mdpi-res.com Internet	10 words — < 1%
74	www.arxiv-vanity.com Internet	10 words — < 1%
75	www.ijraset.com Internet	10 words — < 1%
76	Hari Mohan Rai, Kalyan Chatterjee. "Detection of brain abnormality by a novel Lu-Net deep neural CNN model from MR images", Machine Learning with Applications, 2020 Crossref	9 words — < 1%

77 Md. Farhad Hossain, Md. Ariful Islam, Syed Naimatullah Hussain, Debprosad Das, Ruhul Amin, Mohammad Saydul Alam. "Brain Tumor Classification from MRI Images Using Convolutional Neural Network", 2021 IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAIET), 2021
Crossref

9 words — < 1%

78 Smita Deshmukh, Divya Tiwari. "Chapter 26 Detection and Classification of Brain Tumor Using Convolutional Neural Network (CNN)", Springer Science and Business Media LLC, 2022
Crossref

9 words — < 1%

79 Upasana Kapoor-Narula, Nibedita Lenka. "Elucidating the Anti-Tumorigenic Efficacy of Oltipraz, a Dithiolethione, in Glioblastoma", Cells, 2022
Crossref

9 words — < 1%

80 escholarship.org
Internet

9 words — < 1%

81 etheses.whiterose.ac.uk
Internet

9 words — < 1%

82 www.irjmets.com
Internet

9 words — < 1%

83 Akula Shravya Sri, Bobbili Varshith Reddy, Kanuri Balakrishna, Vollala Akshitha, Sreedhar Kollem, Ch Rajendra Prasad. "Detection Of MRI Brain Tumor Using Customized Deep Learning Method Via Web App", 2023 International Conference on Recent Trends in Electronics and Communication (ICRTEC), 2023
Crossref

8 words — < 1%

84	Md. Alamin Talukder, Md. Manowarul Islam, Md. Ashraf Uddin, Arnisha Akhter et al. "An efficient deep learning model to categorize brain tumor using reconstruction and fine-tuning", Expert Systems with Applications, 2023 Crossref	8 words — < 1%
85	Muthu Krishnammal P, S. Selvakumar Raja. "Deep Learning Based Image Classification and Abnormalities Analysis of MRI Brain Images", 2019 TEQIP III Sponsored International Conference on Microwave Integrated Circuits, Photonics and Wireless Networks (IMICPW), 2019 Crossref	8 words — < 1%
86	Susan Mahler Zneimer. "Cytogenetic Laboratory Management", Wiley, 2016 Crossref	8 words — < 1%
87	Syeda Akter, Lawrence Holder. "Improving IoT Predictions through the Identification of Graphical Features", Sensors, 2019 Crossref	8 words — < 1%
88	Tran, John. "Anatomical Study of the Knee and Shoulder Joint Capsules: Innervation to Innovative Interventions", University of Toronto (Canada), 2020 ProQuest	8 words — < 1%
89	ece.northsouth.edu Internet	8 words — < 1%
90	ijisae.org Internet	8 words — < 1%
91	researchspace.ukzn.ac.za Internet	8 words — < 1%

92	thesai.org Internet	8 words — < 1%
93	www.scilit.net Internet	8 words — < 1%
94	"Signal and Image Processing Techniques for the Development of Intelligent Healthcare Systems", Springer Science and Business Media LLC, 2021 Crossref	7 words — < 1%
95	Geoffrey E. Hinton, Simon Osindero, Yee-Whye Teh. "A Fast Learning Algorithm for Deep Belief Nets", Neural Computation, 2006 Crossref	7 words — < 1%
96	Kaplan Kaplan, Yılmaz Kaya, Melih Kuncan, H. Metin Ertunç. "Brain tumor classification using modified local binary patterns (LBP) feature extraction methods", Medical Hypotheses, 2020 Crossref	7 words — < 1%
97	Marwa M. Emam, Nagwan Abdel Samee, Mona M. Jamjoom, Essam H. Houssein. "Optimized deep learning architecture for brain tumor classification using improved Hunger Games Search Algorithm", Computers in Biology and Medicine, 2023 Crossref	7 words — < 1%
98	Towsif Raiyan, Humayra Hossain Anonna, Sudwipto Kumar Mondal, Mohammad Monirujjaman Khan. "Brain Tumor Detection using Smart Deep Learning", 2022 IEEE 13th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), 2022 Crossref	7 words — < 1%

99 Yakub Bhanothu, Anandhanarayanan Kamalakannan, Govindaraj Rajamanickam. "Detection and Classification of Brain Tumor in MRI Images using Deep Convolutional Network", 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), 2020 — 7 words — < 1%
Crossref

100 Zahid Rasheed, Yong-Kui Ma, Inam Ullah, Tamara Al Shloul et al. "Automated Classification of Brain Tumors from Magnetic Resonance Imaging Using Deep Learning", Brain Sciences, 2023 — 7 words — < 1%
Crossref

101 "Recent Innovations in Computing", Springer Science and Business Media LLC, 2022 — 6 words — < 1%
Crossref

102 Alsahaf, Ahmed Yousif. "Prediction of Flow Zone Indicators in Carbonate Reservoirs Using Hybrid Modeling", King Fahd University of Petroleum and Minerals (Saudi Arabia), 2023 — 6 words — < 1%
ProQuest

103 Manvi Bohra, Siddharth Gupta. "Pre-trained CNN Models and Machine Learning Techniques for Brain Tumor Analysis", 2022 2nd International Conference on Emerging Frontiers in Electrical and Electronic Technologies (ICEFEET), 2022 — 6 words — < 1%
Crossref

104 Ye, Yuyun. "Application of Deep Learning in Tumor Detection", The University of Tulsa, 2023 — 6 words — < 1%
ProQuest

105 ieeexplore.ieee.org — 6 words — < 1%
Internet

106 www.nhs.uk
Internet

6 words — < 1%

EXCLUDE QUOTES	OFF	EXCLUDE SOURCES	OFF
EXCLUDE BIBLIOGRAPHY	ON	EXCLUDE MATCHES	OFF