American International University-Bangladesh (AIUB)

Brain Tumor Detection and Categorization Using Deep Learning Techniques

Rameen Farhan (20-41981-1)

Raqibul Alam (20-42692-1)

MD Minhajul Hoque Chowdhury (20-43086-1)

MD Shahadat Hossen (20-43083-1)

A Thesis submitted for the degree of Bachelor of Science (BSc) in Computer Science and Engineering (CSE) at American International University Bangladesh in February, 2024

Faculty of Science and Technology (FST)

Abstract

Brain tumor classification and detection plays a crucial role in medical diagnosis and treatment planning. In this research, we present a comprehensive approach to enhance the accuracy of brain tumor classification and detection using a multi-class dataset comprising three types of brain tumors along with a class denoting the absence of a tumor. Our methodology integrates various image preprocessing techniques, diverse classifiers, and convolutional neural network (CNN) models to achieve robust classification performance. We first employ Gaussian, Unsharp Mask, and Laplacian filters to enhance the quality and highlight important features in the input images. Subsequently, three pre-trained CNN models - VGG19, ResNet50, and MobileNetV2 - along with a custom-designed CNN architecture are utilized for feature extraction. These models are adept at capturing hierarchical representations of image features. For feature extraction, we adopt a strategy of obtaining 1x1024 feature vectors from the pre-trained CNN models. These extracted features are then fed into three classical classifiers: Random Forest, AdaBoost, and k-Nearest Neighbors (KNN). Through this fusion of preprocessing, deep learning, and classical machine learning techniques, we aim to harness the complementary strengths of each approach for improved classification accuracy. Experimental results on the multi-class brain tumor dataset demonstrate promising outcomes, with an achieved classification accuracy of 97.1%. This high accuracy underscores the effectiveness of our proposed methodology in accurately identifying and categorizing brain tumor types, thereby facilitating more informed medical decisions and treatments. Our study contributes to the ongoing efforts in leveraging advanced computational techniques for enhancing medical image analysis and diagnosis.

Declaration by author

This thesis is composed of our original work, and contains no material previously published or written by another person except where due reference has been made in the text. We have clearly stated the contribution of others to our thesis as a whole, including statistical assistance, survey design, data analysis, significant technical procedures, professional editorial advice, financial support and any other original research work used or reported in our thesis. The content of our thesis is the result of work we have carried out since the commencement of Thesis.

We acknowledge that copyright of all material contained in my thesis resides with the copyright holder(s) of that material. Where appropriate we have obtained copyright permission from the copyright holder to reproduce material in this thesis and have sought permission from co-authors for any jointly authored works included in the thesis.

Rameen Farhan
20-41981-1
Computer Science & Engineering
Faculty of Science & Technology

MD Minhajul Hoque Chowdhury
20-43086-1
Computer Science & Engineering
Faculty of Science & Technology

Raqibul Alam
20-42692-1
Computer Science & Engineering
Faculty of Science & Technology

MD Shahadat Hossen
20-43083-1
Computer Science & Engineering
Faculty of Science & Technology

Approval

The thesis titled **"Brain tumor detection and categorization using deep learning techniques"** has been submitted to the following respected members ofthe board of examiners of the department of computer science in partial fulfilment of the requirementsfor the degree of Bachelor of Science in Computer Science on **(date of defence)** and has been acceptedas satisfactory.

DR. S M Hasan Mahmud
Assistant Professor, Computer Science
Faculty of Science and Technology
American International University-Bangladesh

Sazzad Hossain
Assistant Professor, Computer Science
Faculty of Science and Technology
American International University-Bangladesh

DR. Akinul Islam Jony
Associate Professor & Head (UG)
Faculty of Science and Technology
American International University-Bangladesh

Prof. Dr. Dip Nandi
Associate Dean
Faculty of Science and Technology
American International University-Bangladesh

Mashiour Rahman
Sr. Associate Professor & Dean-in-charge
Faculty of Science and Technology
American International University-Bangladesh

Contributions by authors to the thesis

List the significant and substantial inputs made by different authors to this research, work and writing represented and/or reported in the thesis. These could include significant contributions to: the conception and design of the project; non-routine technical work; analysis and interpretation of researchdata; drafting significant parts of the work or critically revising it so as to contribute to the interpretation.

	Rameen Farhan	**Raqibul Alam**	**MD Minhajul Haque Chowdhuary**	**MD Shahadat Hossen**	**Contribution (%)**
	20-41981-1	*20-42692-1*	*20-43086-1*	*20-43083-1*	
Conceptualization					100 %
Data curation					100 %
Formal analysis					100 %
Investigation					100 %
Methodology					100 %
Implementation					100 %
Validation					100 %
Theoretical derivations					100 %
Preparation of figures					100 %
Writing – original draft					100 %
Writing – review & editing					100 %

If your task breakdown requires further clarification, do so here. Do not exceed a single page.

Acknowledgments

First of all, I would like to thank almighty Allah, for his grace in accomplishing our thesis timely and our families for giving us mental and financial support throughout the whole bachelors.

We would like to express our gratitude to the Faculty of Science & Technology to keep thesis credit in the curriculum of the graduation program and giving us a scope of tasting the flavor or research work with our interest.

We also want to express our gratitude to our supervisor, Dr. SM Hasan Mahmud sir, who gave us motivation, guidance, and control for this venture. He has decent knowledge of programming advancement, and his personal circumstances, perpetual assistance made the task do able.

Keywords

Brain Tumor, MRI, CT, X-Ray, CNN, Pre-trained, Feature Extraction, Random Forest, KNN, Filte.

Table of Contents

List of Figures

List of Tables

List of Abbreviations and Symbols

Mention all the abbreviations and the different symbols that is used in this document.

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 hyperparameter
etc.	*etc.*

Chapter 1

Introduction

1.1 Thesis topic

Brain Tumor Detection and Categorization Using Deep Learning Techniques

1.2 Introduction

Brain tumor classification and detection is paramount in medical imaging for accurate diagnosis and treatment planning. This thesis proposes a comprehensive methodology for brain tumor classification using a diverse dataset encompassing three distinct tumor types along with a tumor-free class. Leveraging advanced image preprocessing filters and state-of-the-art deep learning models, including VGG19, ResNet50, and MobileNetV2, alongside classical machine learning classifiers, such as Random Forest, AdaBoost, and KNN, our approach aims to achieve high accuracy in distinguishing between different tumor types. This research contributes to advancing the field of medical image analysis, with significant implications for enhancing patient care and treatment outcomes.

1.3 Medical Images

Medical imaging plays a pivotal role in modern healthcare, enabling non-invasive visualization of internal structures and facilitating early detection, diagnosis, and treatment monitoring of various medical conditions. Among the diverse modalities employed in medical imaging, such as X-ray, MRI, CT, and ultrasound, the analysis of medical images holds immense potential for advancing diagnostic accuracy and patient care. Particularly in the context of neuroimaging, the intricate structures and abnormalities of the brain necessitate sophisticated image analysis techniques for precise interpretation. In this research, we focus on the domain of medical image analysis, specifically targeting brain tumor classification, a critical task with profound implications for patient management and treatment decision-making. Leveraging a diverse dataset collected from Kaggle, comprising various types of brain tumors alongside tumor-free samples, we aim to harness computational methods and advanced machine learning algorithms to enhance the efficacy and efficiency of medical image analysis. Ultimately, our goal is to contribute to the improvement of healthcare outcomes for individuals affected by brain tumors.

1.4 Motivation

The motivation behind this research stems from the pressing need for improved methods in medical imaging analysis, particularly in the domain of brain tumor classification. Brain tumors present complex challenges in diagnosis and treatment planning, often requiring precise identification and characterization for effective patient management. Existing approaches to brain tumor classification face limitations in accuracy and efficiency, highlighting the necessity for novel methodologies that can address these shortcomings.

Furthermore, the availability of large-scale medical imaging datasets, such as those accessible through platforms like Kaggle, presents an opportunity to leverage advanced computational techniques and machine learning algorithms for enhanced medical image analysis. By harnessing the power of these datasets alongside cutting-edge methodologies, we aim to develop a robust framework for brain tumor classification that can significantly impact clinical practice.

The potential impact of this research is substantial, as accurate and efficient brain tumor classification can lead to earlier detection, more precise treatment planning, and improved patient outcomes. By contributing to the advancement of medical imaging analysis techniques, this research has the potential to make a tangible difference in the lives of individuals affected by brain tumors, ultimately improving their quality of life and prognosis.

1.5 Objective

The primary objective of this research is to develop a comprehensive methodology for brain tumor classification using advanced computational techniques and machine learning algorithms. Specifically, our objectives are as follows:

1. To explore and evaluate various image preprocessing filters, including Gaussian, Unsharp Mask, and Laplacian filters, for enhancing the quality and highlighting relevant features in brain tumor images.
2. To investigate the effectiveness of different pre-trained convolutional neural network (CNN) architectures, including VGG19, ResNet50, and MobileNetV2, in extracting discriminative features from brain tumor images.
3. To propose a feature extraction strategy based on pre-trained CNN models, followed by classification using classical machine learning algorithms such as Random Forest, AdaBoost, and k-Nearest Neighbors (KNN).
4. To evaluate the performance of the proposed methodology on a multi-class brain tumor dataset, measuring classification accuracy and comparing against existing approaches.

5. To assess the clinical relevance and potential impact of the developed methodology in enhancing medical image analysis and facilitating more accurate diagnosis and treatment planning for individuals affected by brain tumors.

1.6 Orientation

The essay's second chapter covers the foundations of machine learning, image processing, and brain tumor identification in addition to previous academics' study on the same topic. The implementation, data preprocessing, augmentation, CNN model train, classification, and a quick summary of the methods used with this model are all covered in the third chapter. The next section contains a representation of the train test portion and the tools and libraries used to build this model. Chapter 4 summarizes the specifics by outlining all of the challenges and contrasting each model's output to determine which accuracy is optimal. We covered the paper and our model's shortcomings in chapter 5. Finally, a few closing thoughts and potential directions for further research are covered in chapter 6.

Chapter 2

Literature review

Because of the complexity of human brains, brain tumor detection is one of the most difficult and challenging undertakings in medical science. Using deep learning and image processing methods, medical professionals and researchers are always striving to enhance the accuracy of brain tumor identification. Because of its high detection accuracy, the application of Convolutional Neural Networks (CNN) and its models in brain tumor detection is expanding. New techniques and algorithms are being introduced by researchers to improve the accuracy of brain tumor identification.

In recent years, deep learning methods such as CNN have demonstrated considerable promise in the diagnosis of brain cancers. By deriving intricate patterns and features from MRI data, CNN models are able to precisely pinpoint the locations of brain tumors. For example, Dunsheng Liu et al. suggested a new model for automatic brain tumor identification utilizing CNN classification called Global Average Pooling Residual Network (G-ResNet). With a 95.00% classification accuracy, the model outperforms earlier models by a large margin. [6]

Ouiza Nait Belaid et al. used inputs of grey level of co-occurrence matrix (GLCM) features images and original photos to introduce a deep learning technique based on pre-trained VGG-16 CNNs for the classification of three types of brain cancers. The experimental results show that the original image has more noticeable and distinguishable features when energy is used as input, with an average accuracy of 96.5% when compared to other combinations of inputs. [7]

In order to distinguish between images of normal brain tissue and images of brain tumors, Ramya Mohan and associates presented the MIDNet18 CNN architecture as an alternative to the VGG16 CNN architecture. The accuracy of the MIDNet18 model was 98.7%. In contrast, the VGG16 model achieved 50% accuracy. [8]

Moreover, several CNN models, including ResNet50, Mobile Net V2, and VGG16, have been employed by researchers to identify brain cancers. An algorithm presented by Zahid Rasheed and his colleagues, for example, was tested against the pre-trained VGG16, VGG19, ResNet50, MobileNet V2, and Inception V3 algorithms using benchmarked data. The results of the trial showed a noteworthy 98.04%

categorization accuracy. [9]

Moreover, deep learning models' computational complexity has decreased because to feature extraction, allowing for quicker processing and improved accuracy. For example, Arpit Kumar and colleagues introduced a novel method using an ensemble classifier based on machine learning that combines augmentation and feature extraction techniques. To precisely identify malignancies, the suggested approach makes use of an optimized fusion vector. This hybrid technique performed exceptionally well, utilizing a modified Resnett50 model and HOG (Histogram of Oriented Gradients) to achieve an 88% detection accuracy. The superiority of the suggested strategy was demonstrated by a comparison of the outcomes with the current methods. [10]

Moreover, Scientists have investigated a variety of classifiers for the extraction of characteristics and their subsequent classification into tumor and non-tumor regions. For example, Tonmoy Hossain et al. used the Sci-kit-learn implementation to use six standard classifiers: SVM, KNN, MLP, Logistic Regression, Naive Bayes, and random Forest. They next tried CNN and got an amazing accuracy of 97.87%, which demonstrated its remarkable capabilities.[11]

One method put out by Aryan Sagar Methil entailed performing experimental study on a dataset made up of tumors that varied in terms of their sizes, forms, textures, and locations. Convolutional neural network technology was used to complete the classification job (CNN). The CNN model demonstrated its very compelling performance with a recall rate of 98.55% on the training set and an astounding 99.73% on the validation set.[12]

Three steps make up the technique that A. Ari et al. proposed: preprocessing, image processing-based tumor region extraction, and tumor classification based on the extreme learning machine local receptive fields (ELM-LRF). The accuracy of classifying cranial MR images in experimental studies is 97.18%. The evaluation results showed that the suggested method's efficacy was higher than that of other recent research in the literature. [13]

D. Fabbri et al. studied 989 axial photos from 191 patients in order to avoid the confusion of three different planes with the same diagnostic in neural networks. Neural networks, both convolutional and totally linked, were used for classification. Within these two groups, more tests were calculated by enhancing the initial 512 512 axial photos. An average five-fold cross-validation of 91.43% for the most highly trained neural network shows the classification accuracy of neural networks trained on axial data.

[14]

A. Rehman suggested a brand-new deep learning-based technique for identifying and categorizing tiny brain cancers. Brain tumors are initially extracted using a 3D CNN architecture, and then the tumors are fed into a CNN model that has been pretrained for feature extraction. For testing and validation, three BraTS datasets from 2015, 2017, and 2018 are employed; their corresponding accuracy rates are 98.32, 96.97, and 92.67%. Comparing the suggested design to existing techniques shows that it achieves a similar level of precision.[15]
A three-stream architecture known as multiscale CNNs was created by Liya Zhao et al. to automatically choose the best top three sales of image sizes and to integrate data from different scales of the surrounding areas. The MICCAI 2013 Multimodal Brain Tumor Image Segmentation Benchmark (BRATS) datasets are utilized for training and assessment. The developed multiscale CNNs framework not only combines multimodal features from T1, T1-enhanced, T2, and FLAIR MRI images, but also multimodal features from T1, T1-enhanced, T2, and FLAIR images. Their approach enhances the resilience and accuracy of brain tumor segmentation when compared to the top two BRATS 2012 and 2013 algorithms and conventional CNNS.[16]

CNN algorithms and artificial intelligence were used in the introduction of a brain tumor categorization framework by A. Deshpande et al. The framework's efficacy was assessed both with and without the use of super-resolution techniques. The remarkable accuracy of 98.14% was attained by combining the ResNet50 architecture with super-resolution. The suggested super-resolution framework, which combines CNN, ResNet50, and the discrete cosine transform (DCT), effectively improves the accuracy of tumor classification, according to experimental results utilizing MRI images. [17]

ResNet-50, CNN, and DNN are three different neural networks that Imran Javaid et al. presented. Ultimately, every deconstructed neural network is assigned to a separate dataset. OTSU segmentation is used to isolate a tumor when an image has been accurately confirmed to be one. The experimental findings show that the ResNet-50 algorithm has a minimal test loss of 0.0269. It also has the greatest F1 score of 1.0, precision of 1.00, and high classification accuracy of 0.996. Numerous tests show that the suggested segmentation for tumor identification is accurate and efficient. [18]

H. Khan et al. suggest a novel method that combines data augmentation, image processing, and a convolutional neural network (CNN) to classify brain MRI scan images as malignant or noncancerous. Using transfer learning, they compare their CNN model with pre-trained VGG-16, ResNet-50, and

Inception-v3 models. Even with a limited dataset, their model outperforms VGG-16 (96%), ResNet-50 (89%), and Inception-v3 (75%), with an astounding accuracy of 100%. In addition, this model is more accurate and efficient than pre-trained models currently in use due to its minimal complexity and processing requirements. [19]

Using CNN models and MRI data, Ahmet Çinar created a technique for diagnosing brain tumors. Using the ResNet50 architecture as the foundational model, they added eight more layers and deleted the final five. The created model's astounding 97.2% accuracy was attained. They also evaluated the effectiveness of many other models, including GoogLeNet, InceptionV3, AlexNet, ResNet50, DenseNet201, and InceptionV3. The best-performing model correctly classified pictures of brain tumors. Previous literature research has supported the method's effectiveness and potential for use in computer-aided systems for brain tumor detection.[20]

A novel method for classifying brain cancers in magnetic resonance imaging (MRI) dubbed CNNBCN (Convolutional Neural Network based on Complex Networks) was introduced by Zhiguan Huang et al. The CNNBC network structure is created using randomly generated graph methods, as opposed to manually built networks, and is subsequently converted into a computable neural network via a network generator. The updated CNNBCN model outperforms other models reported in related studies with an astounding 95.49% classification accuracy for brain tumors. In the experiments, the model also shows less test loss when compared to the ResNet, DenseNet, and Mobile Net models. In addition to producing good results in brain tumor classification, the updated CNNBCN advances neural network design methods.[21]

Yakub Bhanothu et al. propose a deep learning system dubbed Faster R-CNN to address the laborious and prone to error manual evaluation procedure of MRI images for tumor diagnosis. The region proposal network and the classifier network in the algorithm both use the VGG-16 architecture as its foundation. It use the Region Proposal Network (RPN) to effectively identify tumors and identify the regions in which they arise. For gliomas, the algorithm attains an average precision rate of 75.18%, for meningiomas, 89.45%, and for pituitary tumors, 68.18%. The algorithm's efficacy as a performance metric in tumor detection and classification is demonstrated by its mean average precision of 77.60% across all tumor classes. [22]

In their study, A. P. Rahmathunneesa et al. assessed how well four pretrained deep learning networks performed in classifying brain cancers into four classes. The chosen networks—AlexNet, ResNet-50,

and GoogLeNet Inception V3—have already undergone pre-training on the imageNet dataset. The research makes use of MRI pictures of glioma brain tumors and preprocesses the images using data augmentation and skull stripping. Based on parameters like accuracy, precision, recall, F1-score, and training/validation time, the network designs are contrasted. According to the experimental results, AlexNet attains an accuracy of 92.98% in a comparatively short amount of time—just 19 minutes—while ResNet50 achieves a greater accuracy of 96.05% over a longer period of time—roughly 100 minutes. [23]

A unique method focused on exploiting X-ray pictures to diagnose brain tumors and enhance patient treatment planning was presented by Tahia Tazin et al. It investigates the use of convolutional neural networks (CNN) to detect brain cancers with the goal of improving accuracy via transfer learning. For deep feature extraction, pre-trained CNN models such as VGG19, InceptionV3, and MobileNetV2 were used. Performance was assessed using the classification accuracy; MobileNetV2 achieved 92% accuracy, InceptionV3 achieved 91%, and VGG19 achieved 88%. The most accurate model was MobileNetV2, which allowed for early tumor diagnosis prior to the development of physical limitations such as paralysis. [24]

A technique that emphasizes the value of early identification and automated methods for brain tumor analysis is put forth by Divjot Kaur et al. Three feature extraction models—VGG16, VGG19, and Inception v3—are presented. These models provide efficient combination methods for producing precise predictions. Brain tumors are classified as benign or malignant using machine learning classifiers. The results show that the maximum accuracy of 99.4% is achieved when VGG16 and a neural network classifier are combined. The significance of automated techniques in raising the precision and effectiveness of brain tumor diagnosis is emphasized by this study. [25]

According to the literature study, a number of methodologies and approaches, such as feature extraction, deep learning-based approaches, and classifiers that use machine learning and image processing techniques, have been presented for the diagnosis of brain tumors. Nonetheless, there is room to improve these methods' accuracy and potency. In order to detect brain cancers more thoroughly and accurately, this research suggests using multiple filters, CNN models, feature extraction, and classifiers.

Chapter 3

Methods

Image capture, putting the image into the dataset, data augmentation, gaussian smoothing, image normalization, dimensionality reduction, one hot encoding, and classification are some of the processes in the suggested methodology. After the photos are first taken, they are pre-processed utilizing techniques like augmentation, smoothing, normalization, etc. 2072 photos of brain tumors can be accessed online.

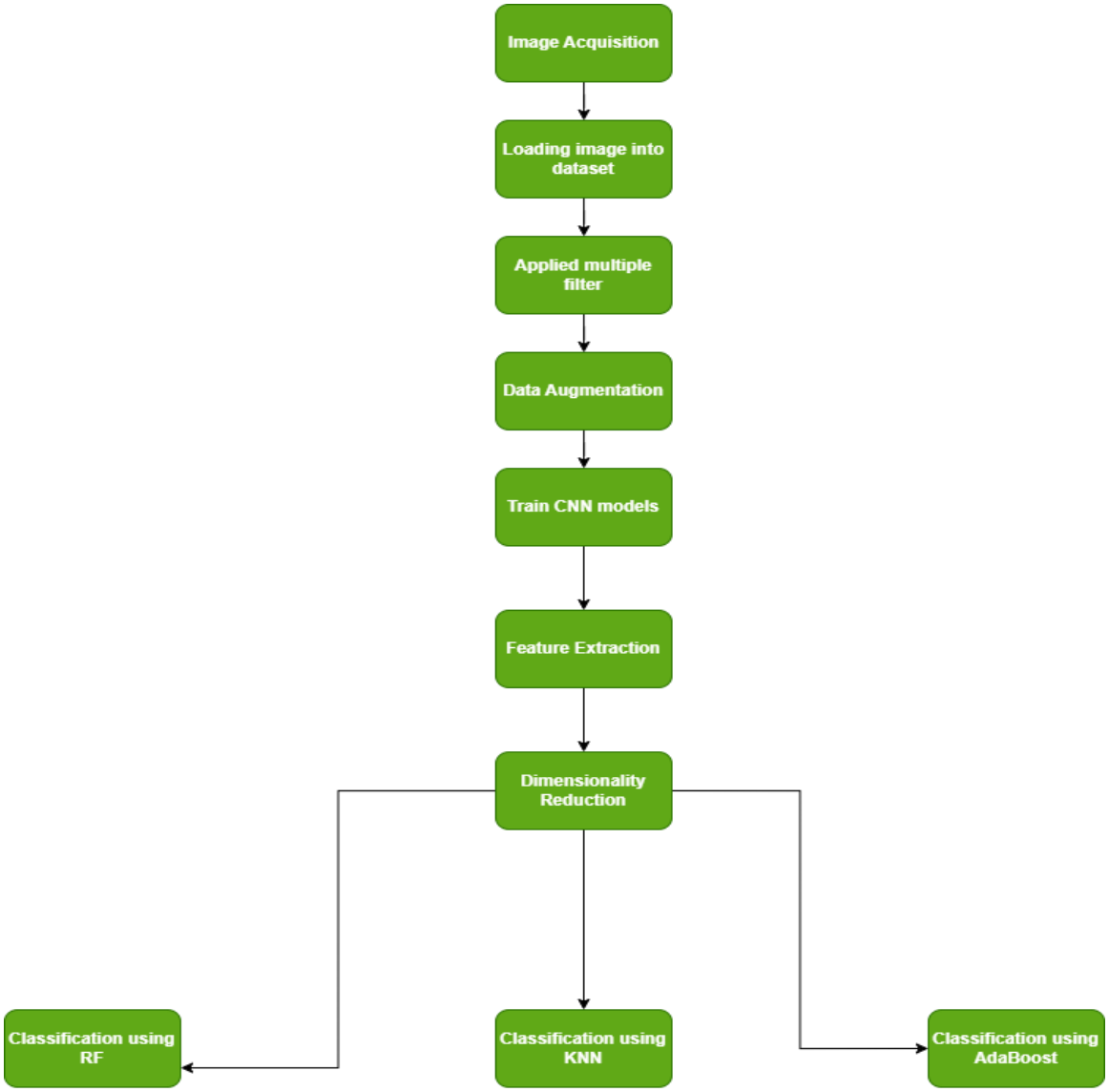

Fig 3.1: Flow chart for proposed methodology

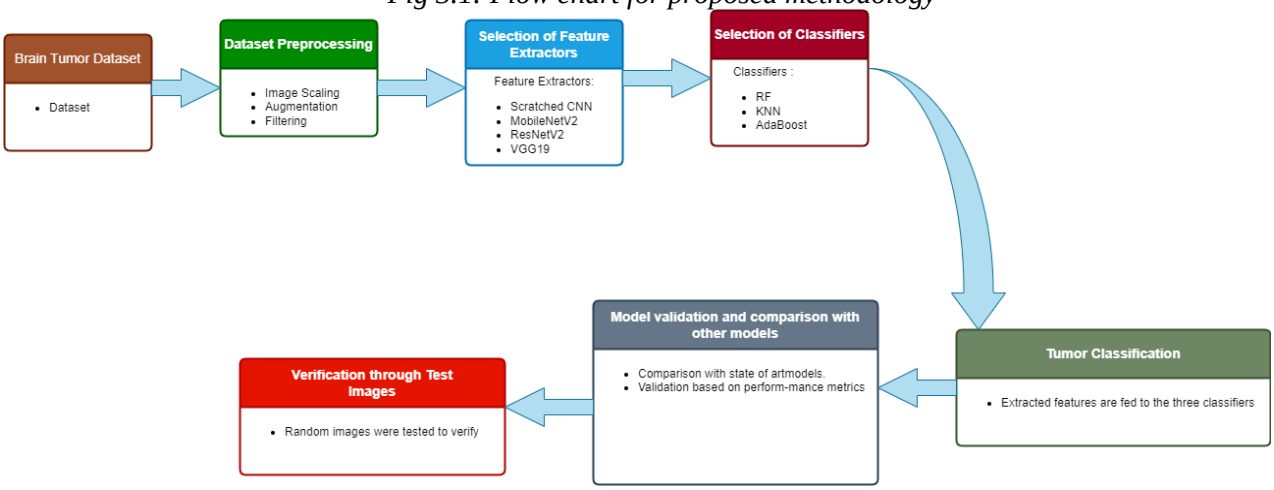

Fig 3.2: Diagram for proposed methodology

3.1 Dataset

A multi-class classification dataset is a type of dataset used in machine learning that assigns

each instance (or data point) to one of several classes. In other words, the aim is to assign data points to more than two groups or classes. This was done using a multi-class classification dataset with four classes. There are gliomas, meningiomas, pituitary tumours, and no tumour.The Glioma, Meningioma, and Pituitary datasets each have 1,168 MRI pictures, whereas the No tumour dataset has 1,175 images. Gaussian, Unsharp mask, and Laplacian filters will be used on these datasets. These four types of datasets will be referred to as Datset 1, Dataset 2, Dataset 3, and Dataset 4, respectively throughout this study.

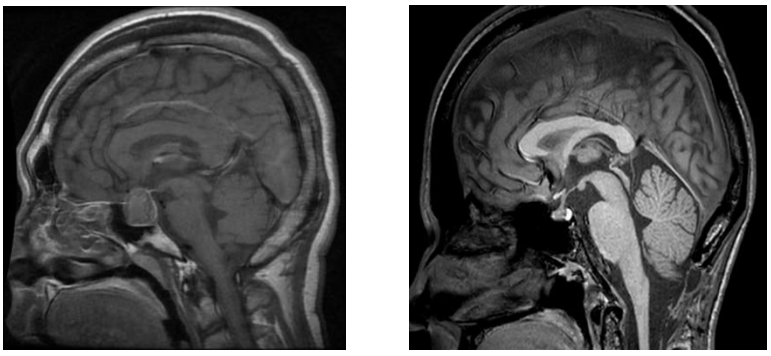

Fig 3.3: Dataset classes

3.2 Data Preprocessing

Data preprocessing is the first phase in data analysis, in which raw data is cleaned, converted, and organized for subsequent analysis. It entails coping with missing numbers, outliers, and guaranteeing data integrity. Data pretreatment may also include scaling, normalization, feature engineering, and dimensionality reduction to prepare the data for modelling. Proper data preprocessing improves the accuracy and efficiency of machine learning algorithms and statistical analysis.

3.2.1 Filtering

3.2.1.1 Unsharp Mask

This dataset contains MRI images applied with the unsharp mask. Unsharp mask filtering is a popular technique used in image processing to enhance image sharpness by accentuating edges and fine details. The process involves subtracting a blurred version of the original image from the original image itself.

Here's how it works:

1. Blurring the Image

The original image is first blurred using a Gaussian filter or another smoothing filter. This blurred image represents the low-frequency components of the original

image.

2. Subtraction

The blurred image is then subtracted from the original image. This process enhances high-frequency components, such as edges and fine details, by emphasizing the differences between neighboring pixel values.

3. Combining with Original Image

The result of the subtraction is then combined with the original image to produce the final sharpened image.

Mathematically, the unsharp mask filtering process can be represented as follows:

Let I be the original image.

Let g be the blurred image obtained by applying a Gaussian filter.

Let S be the sharpened image obtained by subtracting the blurred image from the original image.

The formula for unsharp mask filtering is:

S = I-(K*g)

where k is a scaling factor that controls the strength of the sharpening effect. Typically k is chosen to be between 0.5 and 1.5. Adjusting the value of k allows for fine-tuning the degree of sharpening applied to the image.

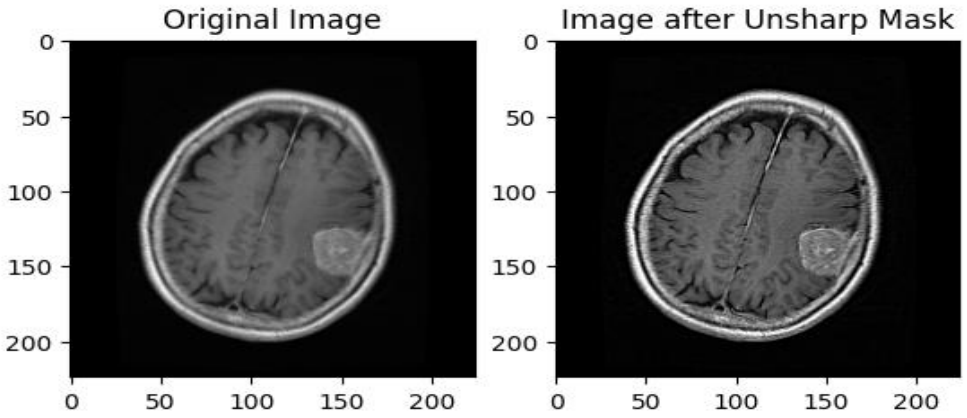

Fig 3.4: unsharp mask filter image

3.2.1.2 Gaussian

This dataset contains MRI images treated using the unsharp mask.Gaussian filtering is an image processing technique that uses a Gaussian function to blur or

smooth a picture. This function calculates a weighted average of neighbouring pixel values, with higher weights allocated to central pixels and decreasing weights to surrounding pixels. The outcome is a smoother image with less high-frequency noise while preserving image borders and details. Gaussian filtering is widely used in computer vision, medical imaging, and photography to reduce noise, denoise images, and perform pre-processing tasks.

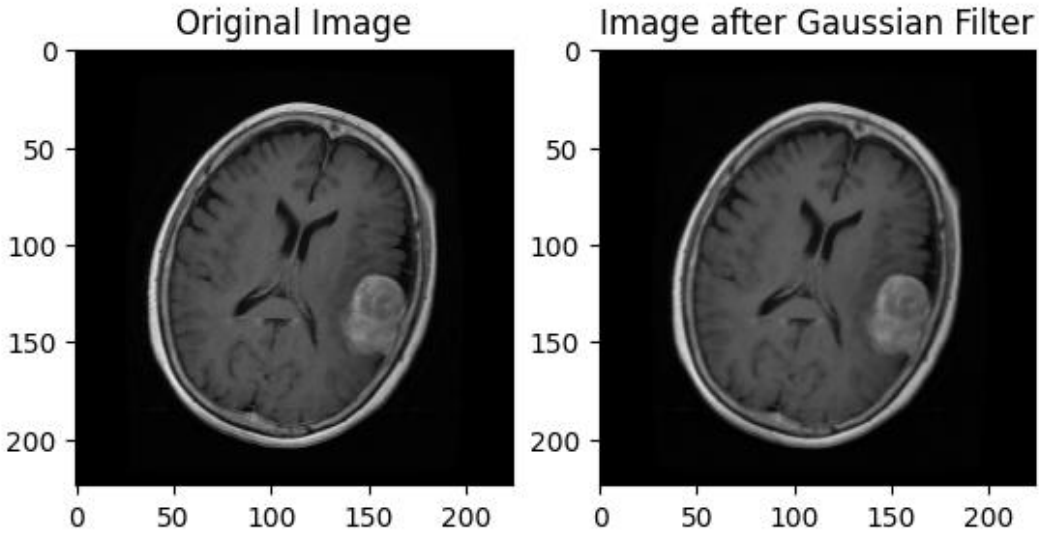

Fig 3.5: Gaussian filter image

3.2.1.3 Laplacian

The Laplacian filter is a technique for image processing that uses the image's second derivative to enhance edges and detect features. It emphasizes places with the highest intensity gradient, such as edges, corners, and boundaries. The filter is created by convolving the image with a Laplacian kernel, which is commonly a 3x3 or 5x5 matrix. This method increases the image's high-frequency components while reducing low-frequency information. The Laplacian filter is widely employed in edge detection, picture sharpening, and feature extraction activities across a variety of domains, including computer vision, medical imaging, and digital image processing. However, it can also enhance noise, necessitating careful parameter tweaking or post-processing procedures to limit the impacts.

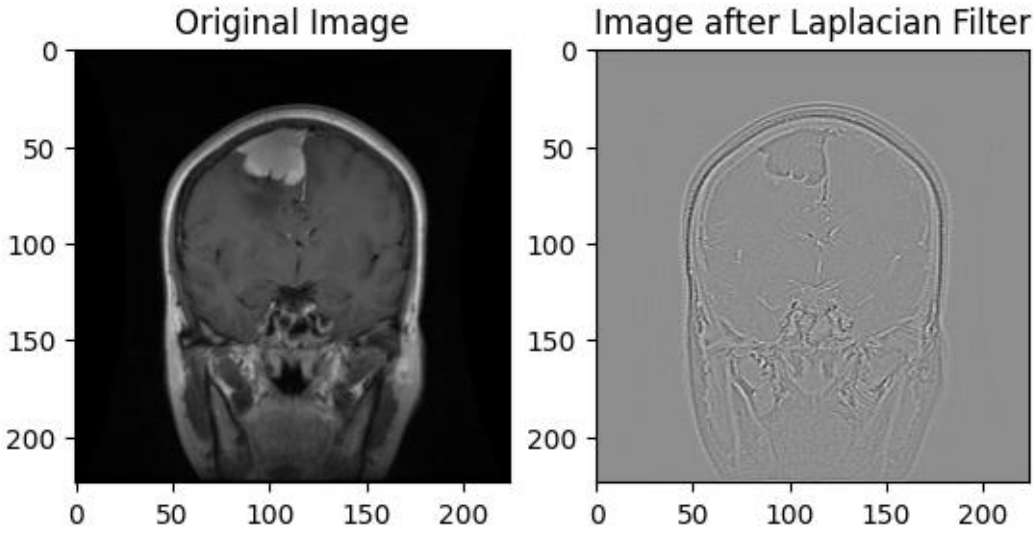

Fig 3.6: Laplacian filter image

3.2.2 Data Augmentation

The datasets undergo minimal preprocessing, including image scaling and augmentation. All photos are scaled to 224*224 pixels. First, the dataset was used as the input picture data. The rescale parameter is then used to normalize the images' pixel values by dividing them by 255. The dataset was then split into validation and training sets with validation_split set to 20%.

Validation Split = Validation size / Total size

In addition, the zoom range was set to 0.99, causing the photos to zoom in and out at random. Following that, the picture data is provided to the preprocessing train function, which accepts a path to a directory holding the training photos and returns an image generator.

zoom_range = (min_zoom , max_zoom)

After that, the image_data object is utilised to create batches of training photos.The target size is set at 224 * 224 pixels.Each batch will contain eight photos. To ensure that the results are reproducible, the seed is set to 123. Finally, the subset parameter is set to "training", indicating that this is the training set.

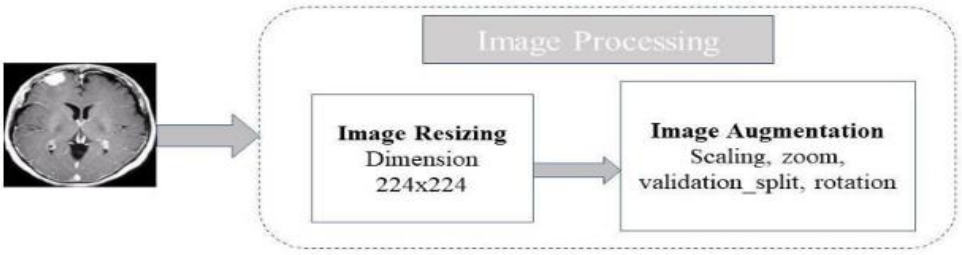

Fig 3.7: Data Processing Stages

3.3 Feature Extraction

Feature extraction is the technique of lowering the dimensionality of data while maintaining useful information. It entails converting raw input data into a feature space, with each dimension representing a significant component or attribute of the data. This technique is widely utilised in machine learning and pattern recognition tasks to increase computational efficiency and model performance.

Three pre-trained models—VGG16, Resnet50, MobilenetV2, and a Scratch CNN model—were investigated for feature extraction. We applied all of the pretrained models listed above to the dataset. We used feature extraction on the trained data, extracting 1*1024 features from each image.

3.3.1 Modified Scratched CNN

Convolutional Neural Networks are a type of neural network that processes data with a grid-like structure, such as pictures or audio spectrograms.CNN learns and extracts information from input images using a combination of convolutional filters, activation, pooling, and fully connected layers.In this study, the CNN architecture was adjusted and polished by making changes to improve its performance. This modified CNN architecture is made up of numerous layers, including a convolutional layer, maximum pooling layers, a flatten layer, and dense layers.

$$Z_i = b_i + \sum_{j=1}^{F} \sum_{k=1}^{F} W_{j,k} X_{i+j,i+k-1} \quad (1)$$

The first layer is an input layer. It captures a picture of the shape (224,224,3). Here, 224*224 denotes height and width, whereas 3 represents the RGB colour channels. The input layer consists of three convoulation layers with 32 and 64 filters that use a 3*3 kernal size and the ReLU activation algorithm. Then, max pooling layers are applied to minimise

the spatial dimensions of the feature.

The Flatten layer flattens the maximum pooling layer's output into a 1D array. After that, the flattened layer is passed through dense layers of 2048 and 1024 units, respectively. To optimise the training process and minimise overfitting, a dropout layer is introduced after each dense layer. The dropout layer rate is 0.3. Finally, the output layer consists of two units and a sigmoid activation function.

$$f(x) = 1/(1+\exp(-x)) \quad (2)$$

This updated CNN architecture incorporates dropout and batch normalisation techniques, which enhances model accuracy and stability.

3.3.2 Modified MobilenetV2

MobileNetV2 is a version of the mobileNet architecture, a form of convolutional neural network intended for mobile and embedded devices with minimal processing resources. Google introduced it in 2018. The input shape is (224,224,3), and the pre-trained layers are not trainable. To prevent overfitting, a 0.3-rate dropout layer is added, followed by a dense layer with three units and a sigmoid activation function. The model is built with the Adam optimizer, sparse categorical cross-entropy as the loss function, and an accuracy metric for evaluation.

$$Loss = -\sum n_{i=1}\, yi * \log \dot{y}_i \quad (3)$$

3.3.3 Modified Resnet50

ResNet50 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.

$$H_p(q) = -\frac{1}{n} \sum_{i=1}^{N} z_i \log(p(z_i)) + (1-z_i)\log(1-p(z_i)) \quad (4)$$

Also, this model uses binary cross-entropy 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.

3.3.4 Modified VGG19

VGG19 is a deep convolutional neural network architecture designed by academics at the University of Oxford in 2014. VGG19 is composed of 19 layers. There are 16 convolutional layers and three fully linked layers. The VGG19 architecture has been fine-tuned through certain modifications. This update makes advantage of transfer learning, with the base model being a pre-trained VGG19 model with ImageNet weights. The final convolutional block and dense layers from the original VGG19 model are deleted, and two new dense layers of 1024 and 2 units are added, respectively. The output of the pre-trained VGG19 model is flattened and fed into the new dense layers. The learning rate of the Adam optimizer is set to 0.001. Categorical cross-entropy

$$H(p,q) = -\sum x\ p(x) \log q\,(x) \qquad 5$$

3.4 Classifier

3.4.1 Random Forest

Random Forest is a method for classification and regression applications.It combines many decision trees to produce more accurate predictions. The programme randomly selects a subset of the dataset's features and then builds numerous decision trees based on them.The final forecast is formed by aggregating the projections of all the trees in the forest, which are typically based on a majority vote for classification or an average value for regression. In the random forest classifier, n_estimators is set to 100, resulting in 100 decision trees, and the random seed value is set to 4020, ensuring that the same results can be obtained if the code is run again with the same random seed. In random forest, the most commonly predicted class is voting. Equation for margin function.

$$f(x) = \arg \max \sum_{j=1}^{j} 1(y=h_j(x)) \qquad 6$$

$$mg(X,Y) = av_k I(h_k(X) = Y) - \max av_k\ I (h_k(X) = j)$$

3.4.2 K-Nearest Neighbors (KNN)

KNN works by identifying the k closest data points in the training set to a given test point, and then using the class or average value of these neighbors to forecast the 39t point. A smaller value of K leads to a more complex model with reduced bias and more variation. A greater K value, on the other hand, results in a simpler model with more bias and less variation. The ideal value of K is often determined using model selection approaches such as cross-validation. The number of neighbors is set to 2, which means that each prediction will take into account two nearest neighbors.

The Minkowski metric specifies the distance metric to utilize for determining the distance between points in the dataset. The Minkowski distance metric (12) will be identical to the Euclidean distance metric. The equation of distant function (13) and h(x) (14) for KNN is:

$$dist\,(x,z) = \left(\sum_{r=1}^{d} |x_r - z_r|^p\right)^{1/p} \quad 7$$

$$dist(x,x') \geq \max\ dist\,(x,x'')$$

$$h(x) = \text{mode}\,(\{y'':(x'',y'') \in S_x\})$$

3.4.3 AdaBoost

The AdaBoost classifier operates by combining numerous weak classifiers into a strong classifier, with each weak classifier focused on a distinct component of the data. AdaBoost iterative 15 is a sequence of weak classifiers on the training data, with each subsequent classifier focusing more on the cases 15 that were misclassified by the preceding classifiers. AdaBoost assigns weights to training samples based on previous misclassifications. The final prediction is then formed by integrating the predictions of all weak classifiers and weighting them according to their accuracy. It iteratively trains a sequence of weak classifiers on the training data [42], with each successive weak classifier emphasizing the data points misclassified by the prior weak classifier .Here, the n_estimator parameter is set to 10 to employ 10 weak classifiers to improve the model's performance, and the random seed value is set to 2020 to ensure that the same results are obtained if the code is run again with the same random seed.

$$H(x) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$$

$$\varepsilon_m = \sum y_i \neq k_m(x_i)\, w_i^{(m)} / \sum_{i=1}^{n} w_i^{(m)}$$

$$\varepsilon_m = \frac{\sum_{n=1}^{N} w_n^m \; l(f_m(x_n).y_n)}{\sum_{n=1}^{N} w_n^m} \qquad 8$$

3.5 Tools and Libraries Used

3.5.1 TensorFlow

TensorFlow is a strong and popular software library that combines artificial intelligence and machine learning. It is open-source and freely downloadable. TensorFlow development is led by the Google Brain team. It is mostly written in C++, but has a Python interface for easy use and interaction with the C++ framework. Python provides a straightforward interface for users to engage with TensorFlow. Python is also used to perform computing, machine learning, and deep learning tasks. To facilitate a variety of jobs. TensorFlow employs symbolic math computations, data flow, and differentiable programming, with a particular emphasis on deep neural network training. The TensorFlow code is written in Python. However, when developing new techniques or features, C++ is frequently used. TensorFlow is used in different deep learning applications. It provides pre-built structures for deep learning and machine learning methods, such as convolutional neural networks used in computer vision and natural language processing. Image processing and neural networks are used to perform tasks such as sequence labelling, classification, and prediction. This open-source machine learning platform offers comprehensive support, from training to deployment. TensorFlow can be used for various tasks, including handwritten digit classification, image recognition, word embedding, recurrent neural networks, machine translation with sequence models, and natural language processing. TensorFlow allows for a smooth transition from training to large-scale production by employing the same models.It is a versatile framework that empowers researchers and scientists to tackle challenging machine learning tasks, features, and convenience of use via the Python interface.It is a versatile framework that empowers researchers and scientists to tackle challenging machine learning tasks, features, and convenience of use via the Python interface.

3.5.2 Kaggle

Kaggle is a well-known platform that provides easy access to materials and datasets related to machine learning and data science. It is very popular for deep learning applications such as neural networks. Kaggle provides users with a variety of functions for discovering and exploring datasets, as well as building models, all within a website. It also enables collaboration with other data scientists and machine learning engineers to address the issues of this domain. Competitions are frequently conducted on the website, allowing users and participants to exhibit their abilities.Kaggle's dataset benefits us by giving excellent materials and resources, resulting in an engaged community. Kaggle is the ideal venue for beginners looking to learn new skills and start a project, as well as experienced individuals looking to participate in competitive events.The platform covers a variety of disciplines, including Python programming, machine learning, data visualisation, SQL, deep learning, natural language processing (NLP), and image processing. It includes detailed explanations and assistance with these issues, making it a valuable and instructive resource for anybody interested in this field.

3.5.3 Scikit Learn

Scikit-learn is a powerful machine learning toolkit extensively used in Python. It builds on famous Python libraries like NumPy and SciPy. It includes a wide range of techniques for classification, regression, clustering, supervised, and unsupervised machine learning problems. It is built on the Python interface and provides access to a variety of efficient machine learning tools such as classification, regression, clustering, and dimensionality reduction approaches.Scikit-learn simply interfaces with popular Python libraries such as NumPy, SciPy, and Matplotlib. It contains a number of machine-learning methods, including logistic regression, support vector machines (SVM), and random forest.Data scientists can use these methods to do classification, regression, and clustering with ease.Users get access to a wide range of features for examining and analysing data, feature engineering, model training, and evaluation. The library is very user-friendly. It has a consistent, user-friendly design, making it suitable for both new and seasoned data scientists and machine learning practitioners. Scikit-learn is a popular Python machine learning package noted for its ease of use and durability. It enables the user to perform a variety of machine learning tasks rapidly and successfully.

3.5.4 OpenCV

OpenCV is a sophisticated and popular toolkit for real-time computer vision, machine learning, and image processing tasks. It is extremely important in the image processing industry. OpenCV offers a wide variety of tools and functions for image processing and computer vision. OpenCV can process pictures and movies to detect objects and faces.This library provides a wide range of functions, including face identification, object tracking, landmark detection, and much more. It supports several programming languages, including Python, Java, and C++.OpenCV offers a comprehensive library of content for academics and students studying computer vision and image processing.It's extensively used to analyse and process photos.It allows for picture rotation at arbitrary angles, downscaling, and smoothing using filters like Gaussian blur, improving processing efficiency. The library provides efficient and optimised methods for real-time picture processing. OpenCV enables developers and academics to easily deploy advanced computer vision techniques and algorithms. It's commonly utilised in object identification, augmented reality, robotics, and surveillance systems.

3.5.5 Matplotlib

Matplotlib is a popular and capable Python data visualisation library that provides a variety of high-quality plots and charts. It offers a versatile and user-friendly interface for building a variety of visualisations that effectively communicate data insights.

Matplotlib allows us to construct line plots, scatter plots, bar plots, histograms, pie charts, and other types of graphics. It includes a number of customisation options, allowing you to fine-tune every aspect of plots. Colours, line styles, markers, axis labels, titles, legends, and other visual components can be tailored to the user's exact requirements.It interfaces with other Python libraries, like NumPy and Pandas, making it simple to visualise data arrays and execute data analysis tasks. It also integrates with Jupyter Notebook, allowing for interactive graphing and exploration.Matplotlib is used extensively in a variety of fields, including data science, machine learning, scientific research, and data visualisation. It allows users to see patterns, trends, distributions, and linkages in the data. It facilitates data exploration, analysis, and display. The library offers a variety of output formats, including interactive charts and static images for publishing. It also allows you to create animated visualisations, which are ideal for presentations and web applications. Matplotlib is a useful and necessary tool for making visually appealing and informative plots and charts with Python. Its diverse plot styles, customisation options, and integration

make it an invaluable tool for data visualisation and analysis.

3.6 Train Test Split

In deep learning model training, it is usual practice to divide the dataset into three sections: training, testing, and validation. Typically, when we acquire a primary dataset, it is delivered as a single directory or file. To work effectively with the dataset, we must preprocess it and divide it into appropriate subgroups for training and testing. In many circumstances, partitioning the dataset into training and testing subsets is necessary, and the scikit-learn module provides a useful method for this. This function is typically used to divide arrays or datasets into distinct subsets for training and testing. The function in scikit-learn accepts numerous parameters, including the arrays or datasets to be split, the desired test size, and randomization options. Calling this method and giving the required arguments allows the dataset to be randomly divided into training and testing subgroups. This splitting procedure ensures that the model is trained on a subset of the data and then tested on previously unknown data to determine its performance. The validation component is frequently treated separately, and it can be built from training data or combined with testing data to fine-tune and validate the model.by taking this technique, we may properly prepare the dataset for deep learning model training, ensuring that the model is trained on a meaningful fraction of the data and tested on independent data to check its generalisation capabilities.We used 80% of the data for training. We had 3738 entries in our training subset. We allotted 20% of the data for testing, with the test subsets containing approximately 935 items. When the primary data is not originally partitioned into training and testing sections, it is usual practice to train models using the full dataset, which frequently results in overfitting. Overfitting occurs when the model becomes too specialised in learning the training data and fails to generalise well to new data to circumvent this problem, divide the data into separate subsets for training, testing, and validation.The training data is used to train the model, the testing data to assess the model's performance on previously unknown data, and the validation data to fine-tune the model and make decisions on hyperparameter tuning or model selection.It is critical to note that tampering with testing data is extremely unethical and can result in biassed and misleading results. Testing data should be kept separate and not used for model development.

3.7 Evaluation Metrics

3.7.1 Precision

The precision metric is used to assess the performance of classification models. It is determined by dividing the total number of true positive samples by the sum of true positive and false positive samples. Precision refers to how reliably the model recognises positive samples among those it predicted as positive. Precision can also refer to the degree of correctness or agreement between two or more measurements. It estimates the degree of similarity or agreement among various measurements or values. Precision refers to the degree of exactness or accuracy with which a specific dimension or number is determined or measured. It displays how closely related or comparable two or more measures are to each other, demonstrating the degree of precision in the measuring process.

$$\text{Pr}\,e\,cision = \frac{Tp}{Tp + Fp} \quad (9)$$

3.7.2 Accuracy

The accuracy measure is often used to assess the performance of classification models. However, accuracy may not be the best statistic in many situations, particularly when working with datasets with imbalanced classes. In such instances, a classifier that predicts the majority class for each instance can achieve high accuracy while failing to capture the minority class. Our dataset has several classes, which may lead to inaccurate results due to class imbalances.

$$Accuracy = \frac{Tp + Tn}{Tp + Tn + Fp + Fn} \quad (10)$$

3.7.3 Recall

Recall is a key evaluation statistic for appropriately detecting positive events. It's also referred to as sensitivity or true positive rate. Recall assesses a classifier's accuracy at identifying all positive instances in a dataset. True positives (TP) are situations that are accurately assessed as positive by the classifier. False negative refers to events that are truly positive but are wrongly labelled as negative by the classifier.

$$Recall = \frac{(Number\ of\ true\ positives)}{(Number\ of\ true\ positives + Number\ of\ false\ negatives)} \quad 11$$

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 presents the best possible F1 score and 0 indicates the worst F1 score.

$$F1\ Score = \frac{2 \times (precision \times Recall)}{(\ \mathrm{Pr}\, e\ cision\ +\ Recall)} \quad 12$$

Chapter 4

Results or findings

4.1 Analysis of CNN models for original dataset

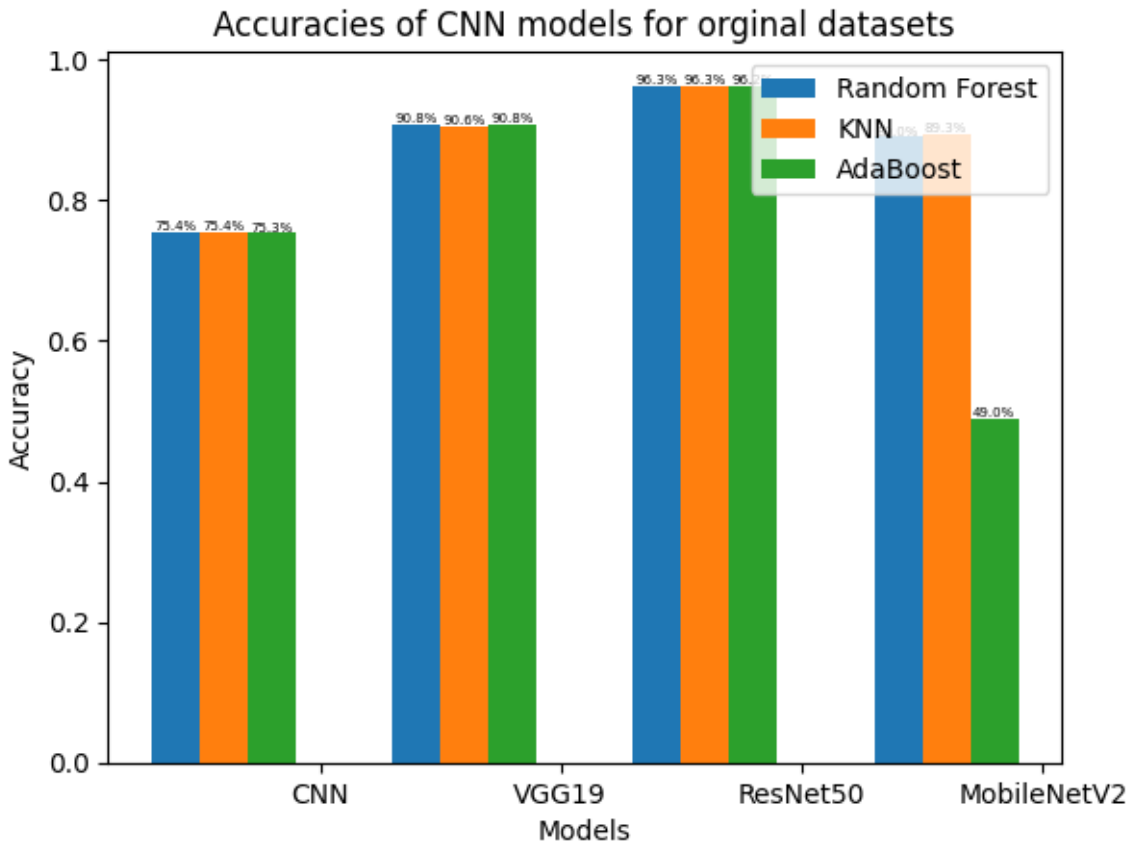

Fig 4.1: Histogram of the accuracy of CNN models on original dataset

For the Scratched CNN model, the Random Forest and KNN classifiers achieved an accuracy of 75.4%, while AdaBoost closely followed with an accuracy of 75.3%. Moving on to the MobileNetV2 model, we observed significantly higher accuracies across all classifiers, with Random Forest achieving 90.8%, KNN at 90.6%, and AdaBoost at 90.8%. The ResNet50 model displayed remarkable accuracy, with all three classifiers achieving an impressive 96.3%. Lastly, for the VGG19 model, Random Forest and KNN both reached accuracies of 89%, while AdaBoost lagged behind with an accuracy of 49%.

4.1.1 Performance Result

Model	Accuracy	Loss	Validation Accuracy	Validation Loss
Scratched CNN	1.0000	5.5317e-06	0.7539	2.8737
MobileNetV2	0.9951	0.0175	0.8919	0.5731
ResNet50	1.0000	1.7019e-04	0.9631	0.1485
VGG19	0.9933	0.0250	0.9080	0.7827

Table 4.1: Accuracy results for original datasets.

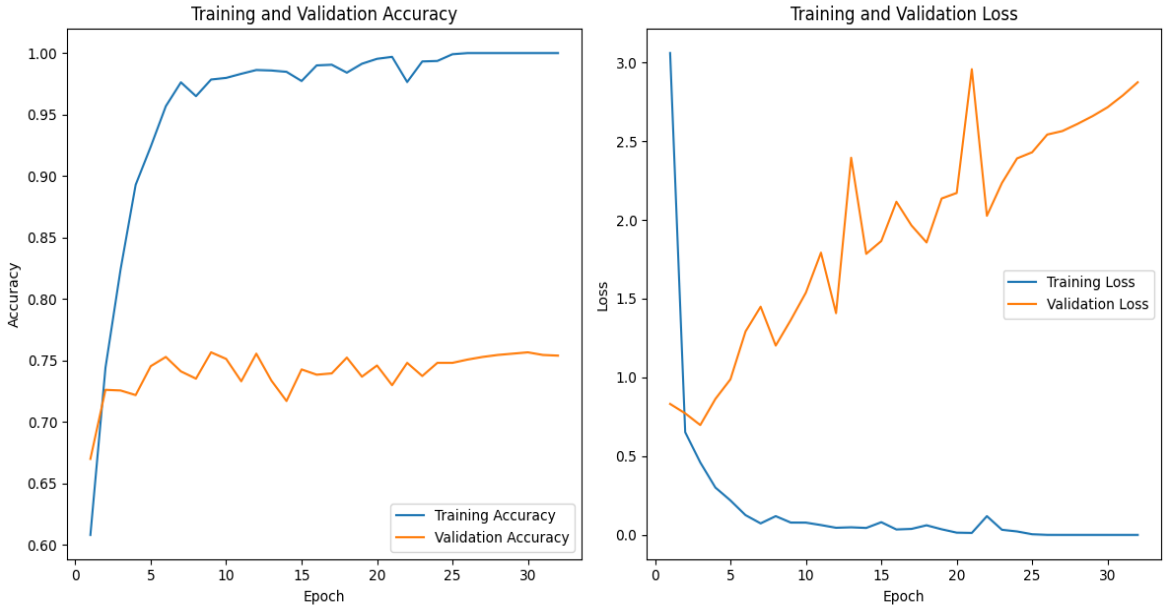

Fig 4.2: Performance Result of Scratched CNN

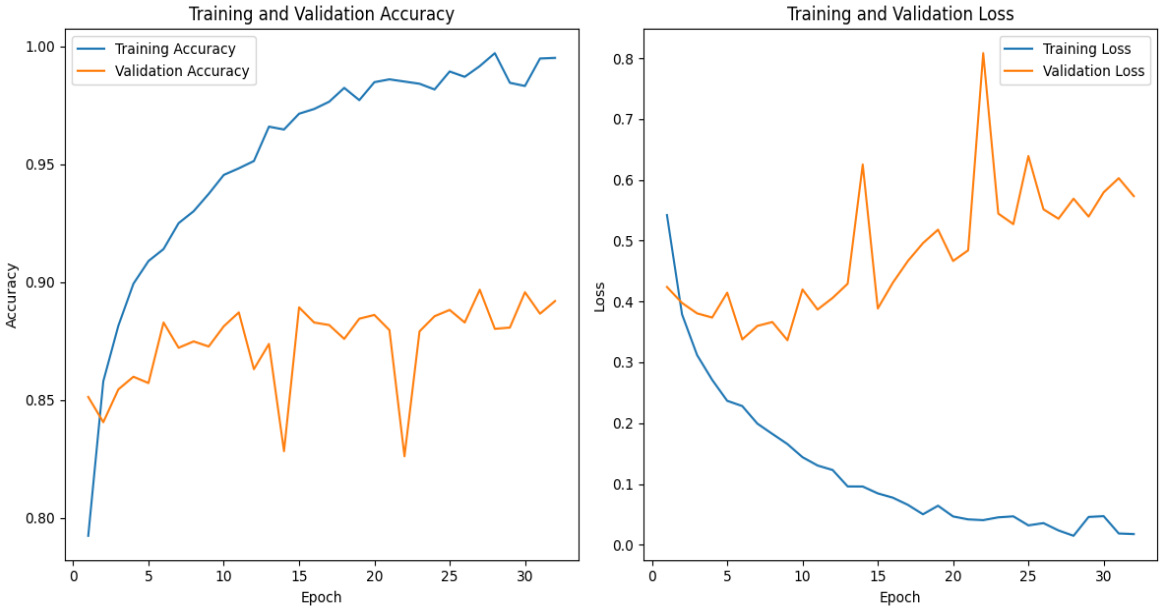

Fig 4.3: Performance Result of mobilenetV2

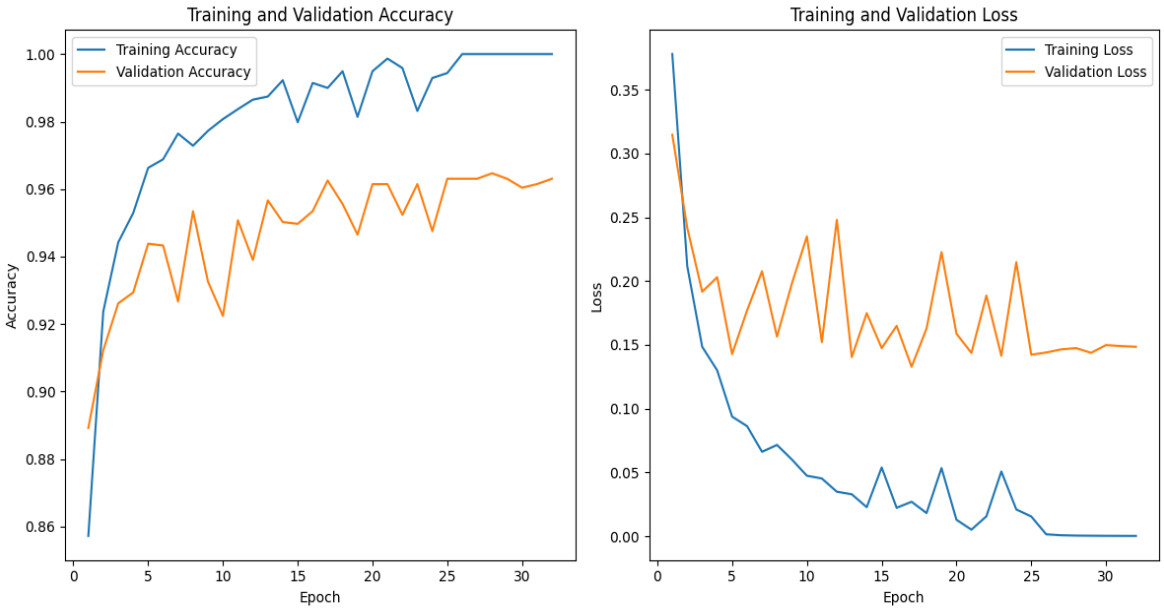

Fig 4.4: Performance Result of Resnet50

Fig 4.5: Performance Result of VGG19

4.1.2 Performance Metrics

Models	Precision	Recall	F1-Score
Scratched CNN	0.7551	0.7539	0.7544
MobileNetV2	0.8905	0.8903	0.8903
ResNet50	0.9632	0.9631	0.9631
VGG19	0.9087	0.9085	0.9084

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

Models	Precision	Recall	F1-Score
Scratched CNN	0.7553	0.7539	0.7545
MobileNetV2	0.8937	0.8935	0.8935
ResNet50	0.9632	0.9631	0.9631
VGG19	0.9055	0.9058	0.9055

Table 4.3: Precision, Recall, F-1 Score for KNN

Models	Precision	Recall	F1-Score
Scratched CNN	0.7548	0.7533	0.7539
MobileNetV2	0.4108	0.4896	0.3655
ResNet50	0.9621	0.9620	0.9620
VGG19	0.9085	0.9085	0.9081

Table 4.4: Precision, Recall, F-1 Score for AdaBoost

4.2 Analysis of CNN models for gaussian filter

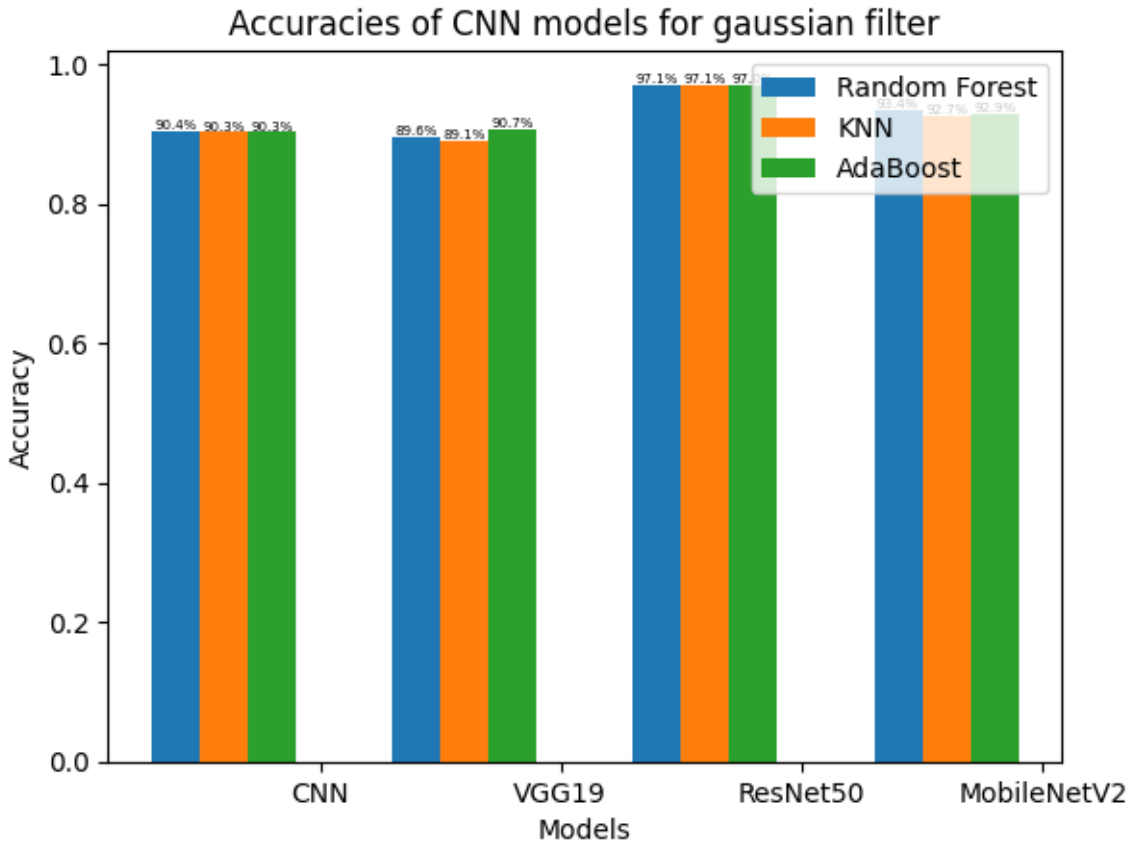

Fig 4.6: Histogram of the accuracy of CNN models on gaussian dataset

For the Scratched CNN model, Random Forest, KNN, and AdaBoost exhibited strong performances, achieving accuracies of 90.4%, 90.3%, and 90.3%, respectively. MobileNetV2 displayed consistent results across classifiers, with Random Forest at 89.6%, KNN at 89.1%, and AdaBoost at 90.7%. ResNet50 emerged as the standout model with remarkable accuracy, where all three classifiers achieved an impressive 97.1%. VGG19, while showcasing commendable accuracy, presented slight variations among classifiers, with Random Forest at 93.4%, KNN at 92.7%, and AdaBoost at 92.9%

4.2.1 Performance Result

Model	Accuracy	Loss	Validation Accuracy	Validation Loss
Scratched CNN	1.0000	3.0788e-06	0.9027	0.9513
MobileNetV2	0.9990	0.0057	0.9308	0.2995
ResNet50	1.0000	3.3650e-04	0.9709	0.1122
VGG19	0.9862	0.0687	0.9057	0.6382

Table 4.5: Accuracy results for gaussian filter datasets.

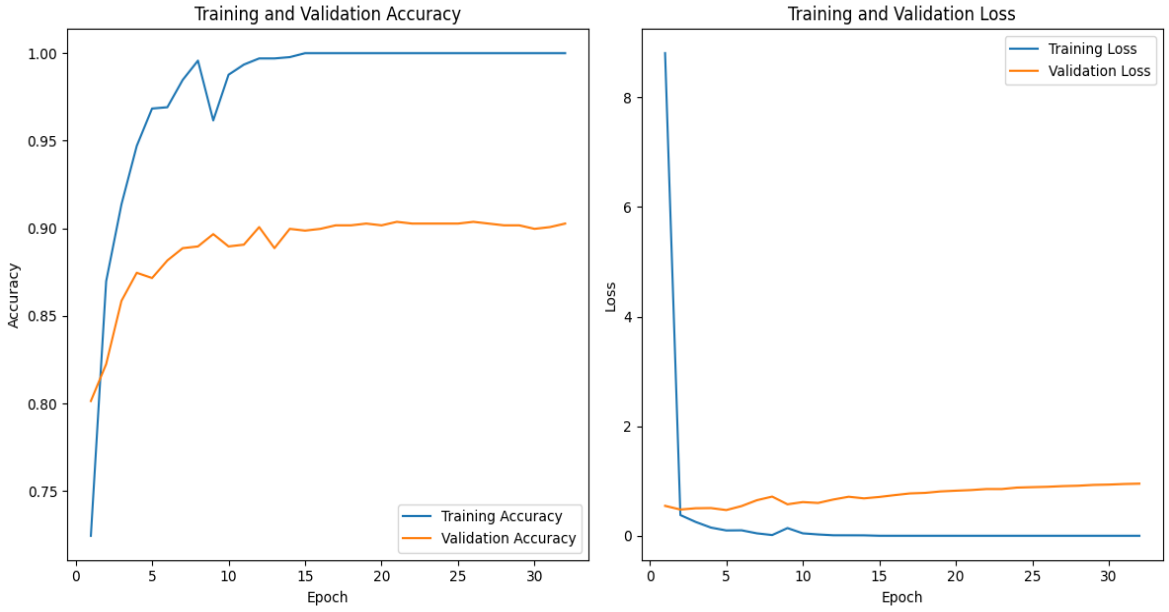

Fig 4.7: Performance Result of Scratched CNN

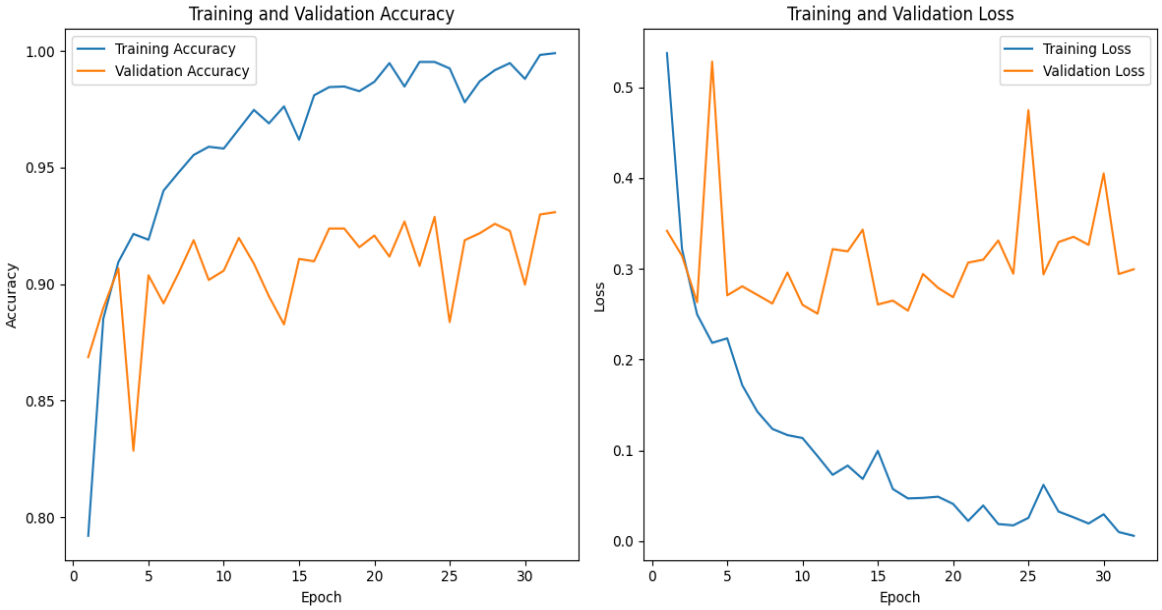

Fig 4.8: Performance Result of mobilenetV2

Fig 4.9: Performance Result of Resnet50

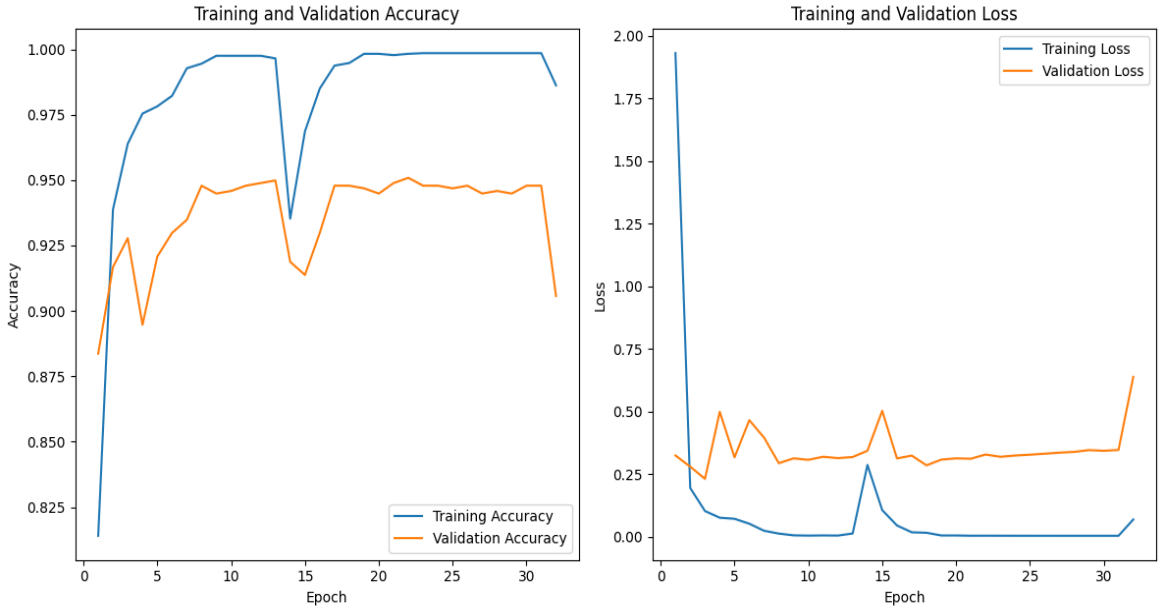

Fig 4.10: Performance Result of VGG19

4.2.2 Performance Metrics

Models	Precision	Recall	F1-Score
Scratched CNN	0.9038	0.9037	0.9037
MobileNetV2	0.9345	0.9338	0.9338
ResNet50	0.9709	0.9709	0.9709
VGG19	0.8951	0.8957	0.8952

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

Models	Precision	Recall	F1-Score
Scratched CNN	0.9029	0.9027	0.9027
MobileNetV2	0.9265	0.9268	0.9263
ResNet50	0.9710	0.9709	0.9709
VGG19	0.8901	0.8907	0.8901

Table 4.7: Precision, Recall, F-1 Score for KNN

Models	Precision	Recall	F1-Score
Scratched CNN	0.9028	0.9027	0.9027
MobileNetV2	0.9267	0.9288	0.9288
ResNet50	0.9699	0.9699	0.9699
VGG19	0.9062	0.9067	0.9062

Table 4.8: Precision, Recall, F-1 Score for AdaBoost

4.3 Analysis of CNN models for Laplacian filter

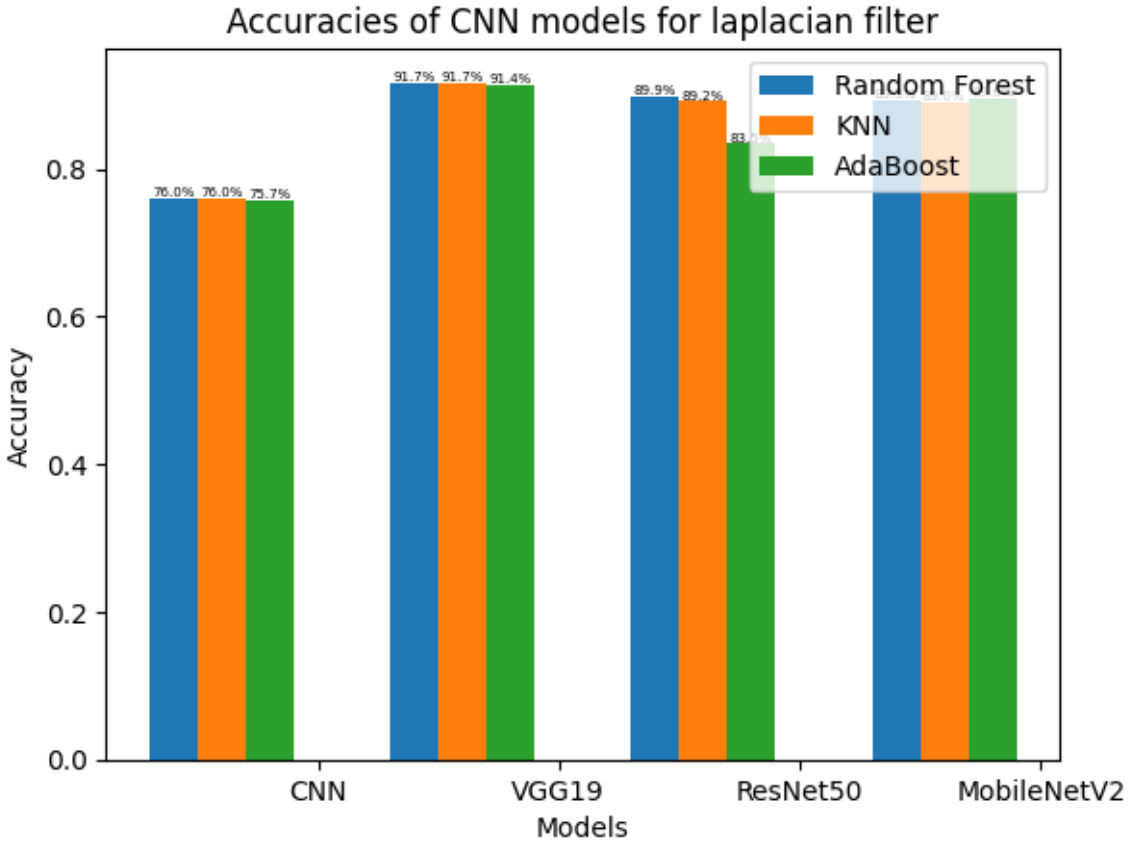

Fig 4.11: Histogram of the accuracy of CNN models on Laplacian dataset

For the Scratched CNN model, all three classifiers demonstrated competitive accuracy rates, with Random Forest and KNN both reaching 76%, and AdaBoost closely trailing at 75.7%. In contrast, MobileNetV2 showcased superior accuracy across the board, with Random Forest, KNN, and AdaBoost achieving impressive accuracies of 91.7%, 91.7%, and 91.4%, respectively. ResNet50, while exhibiting robust accuracy with Random Forest and KNN at 89.9% and 89.2%, faced a slight dip with AdaBoost, landing at 83.4%. Lastly, the VGG19

model demonstrated consistent performance among classifiers, with Random Forest, KNN, and AdaBoost attaining accuracies of 89.3%, 89%, and 89.6%, respectively.

4.3.1 Performance Result

Model	Accuracy	Loss	Validation Accuracy	Validation Loss
Scratched CNN	0.9588	0.1217	0.7623	3.2778
MobileNetV2	0.9719	0.0841	0.8917	0.4385
ResNet50	0.9573	0.1218	0.8806	0.3916
VGG19	0.9859	0.0365	0.9107	0.7806

Table 4.9: Accuracy results for Laplacian filter datasets.

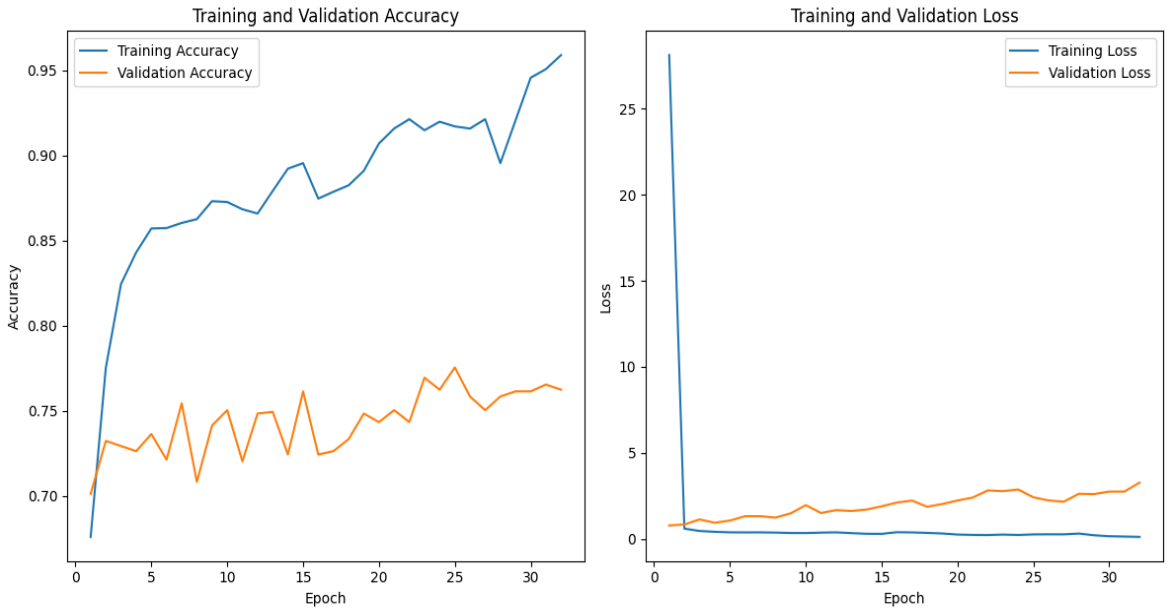

Fig 4.12: Performance Result of Scratched CNN

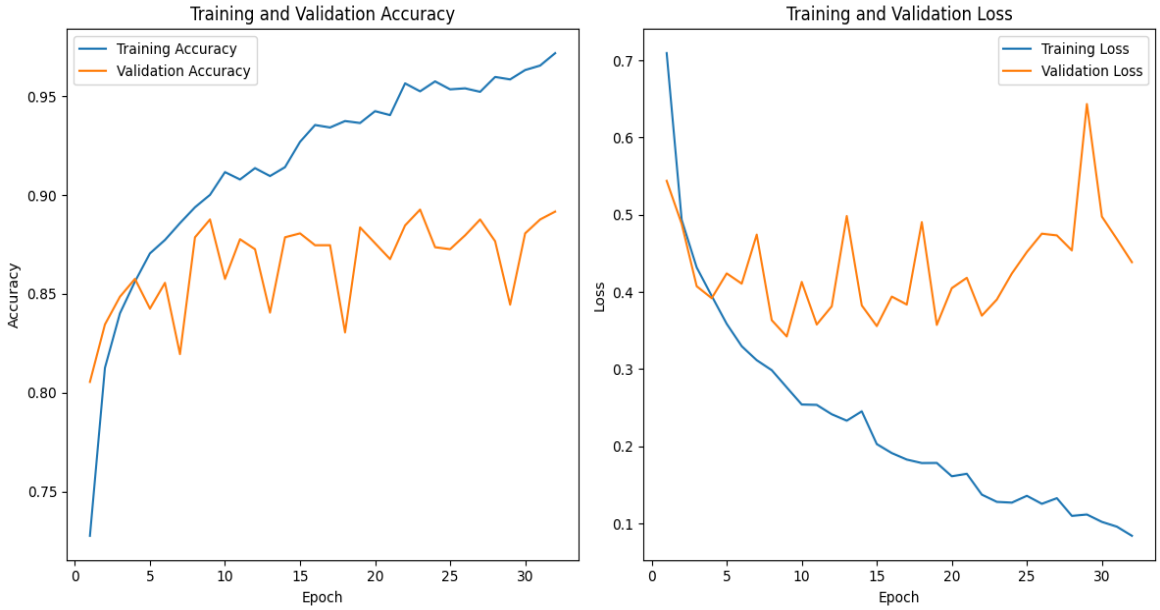

Fig 4.13: Performance Result of mobilenetV2

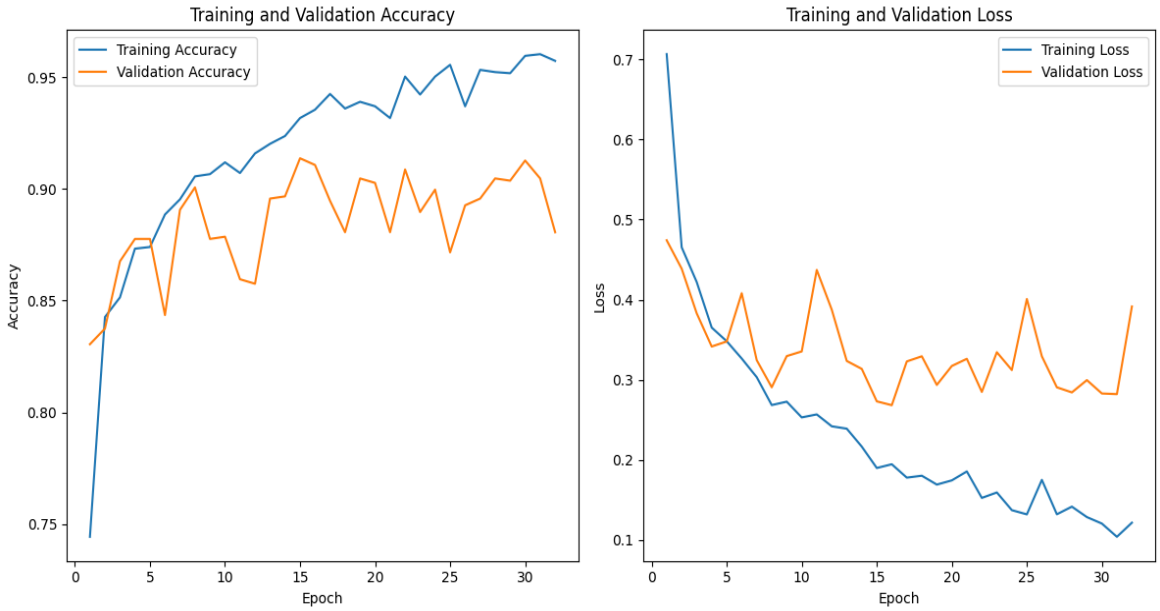

Fig 4.14: Performance Result of Resnet50

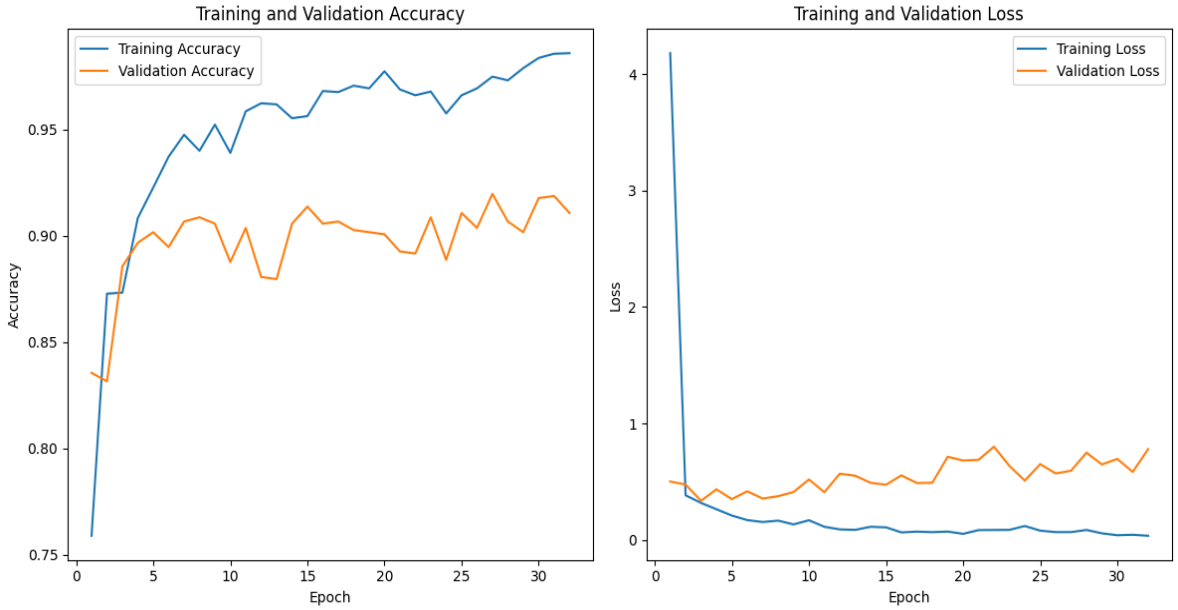

Fig 4.15: Performance Result of VGG19

4.3.2 Performance Metrics

Models	Precision	Recall	F1-Score
Scratched CNN	0.7628	0.7603	0.7610
MobileNetV2	0.8925	0.8927	0.8924
ResNet50	0.8994	0.8987	0.8989
VGG19	0.9170	0.9168	0.9166

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

Models	Precision	Recall	F1-Score
Scratched CNN	0.7618	0.7603	0.7605
MobileNetV2	0.8892	0.8897	0.8892
ResNet50	0.8927	0.8917	0.8920
VGG19	0.9169	0.9168	0.9165

Table 4.11: Precision, Recall, F-1 Score for KNN

Models	Precision	Recall	F1-Score
Scratched CNN	0.7582	0.7573	0.7572
MobileNetV2	0.8952	0.8957	0.8945
ResNet50	0.8591	0.8345	0.8373
VGG19	0.9146	0.9137	0.9129

Table 4.12: Precision, Recall, F-1 Score for AdaBoost

4.4 Analysis of CNN models for unsharp mask filter

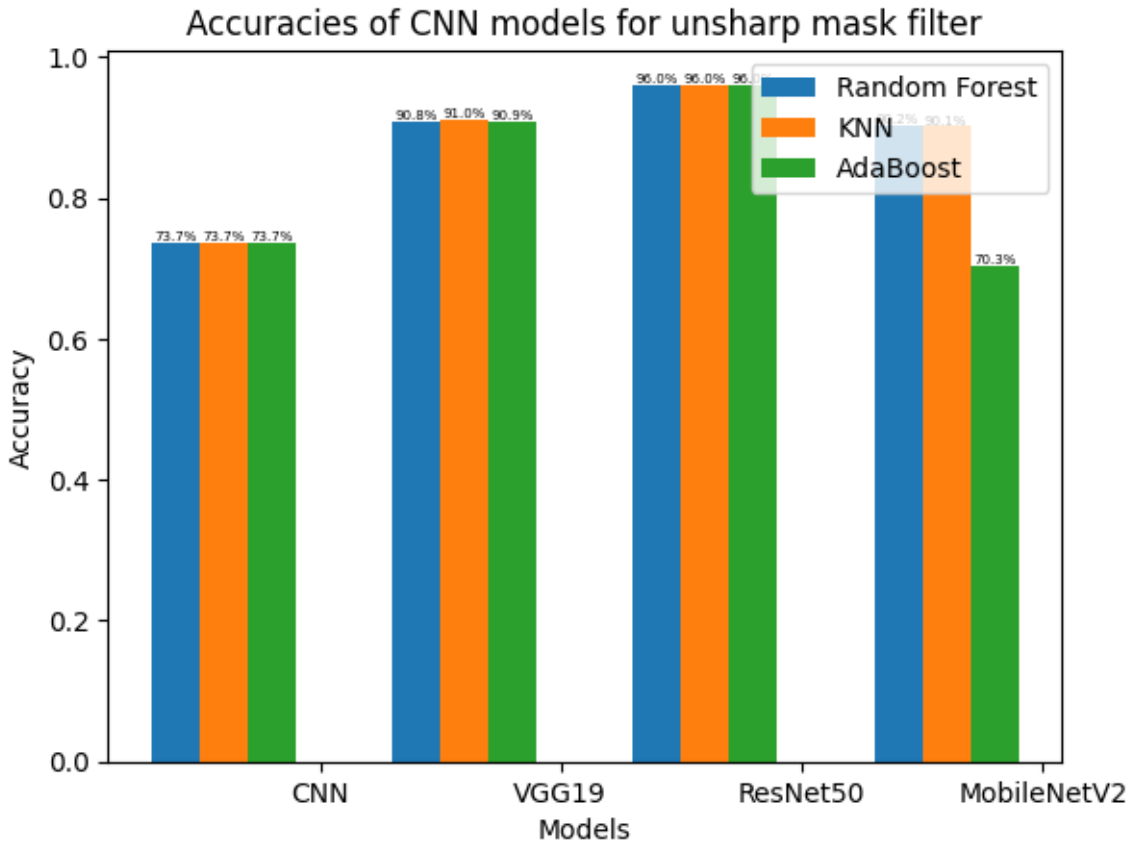

Fig 4.16: Histogram of the accuracy of CNN models on unsharp mask dataset

For the Scratched CNN model, all three classifiers exhibited consistent accuracies, with Random Forest, KNN, and AdaBoost achieving a solid 73.7%. MobileNetV2 showcased strong capabilities in handling the Unsharp Mask Filter dataset, with Random Forest, KNN, and AdaBoost achieving accuracies of 90.8%, 91%, and 90.9%, respectively. ResNet50 demonstrated exceptional performance across all classifiers, boasting an impressive accuracy rate of 96%. In contrast, VGG19 faced some challenges, particularly with AdaBoost, resulting in a lower accuracy of 70.3%, while Random Forest and KNN held steady at 90.2% and

90.1%, respectively.

4.4.1 Performance Result

Model	Accuracy	Loss	Validation Accuracy	Validation Loss
Scratched CNN	1.0000	1.8084e-06	0.7373	3.0253
MobileNetV2	0.9951	0.0163	0.9006	0.4511
ResNet50	1.0000	2.0886e-04	0.9605	0.1830
VGG19	0.9984	0.0049	0.9113	0.8677

Table 4.13: Accuracy results for unsharp mask filter datasets.

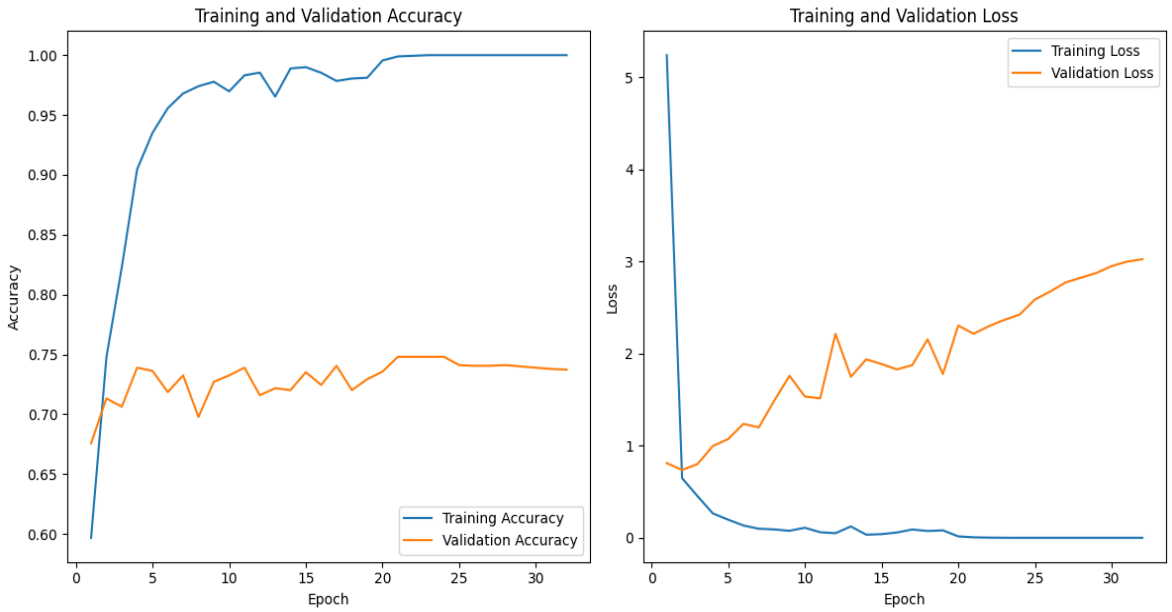

Fig 4.17: Performance Result of Scratched CNN

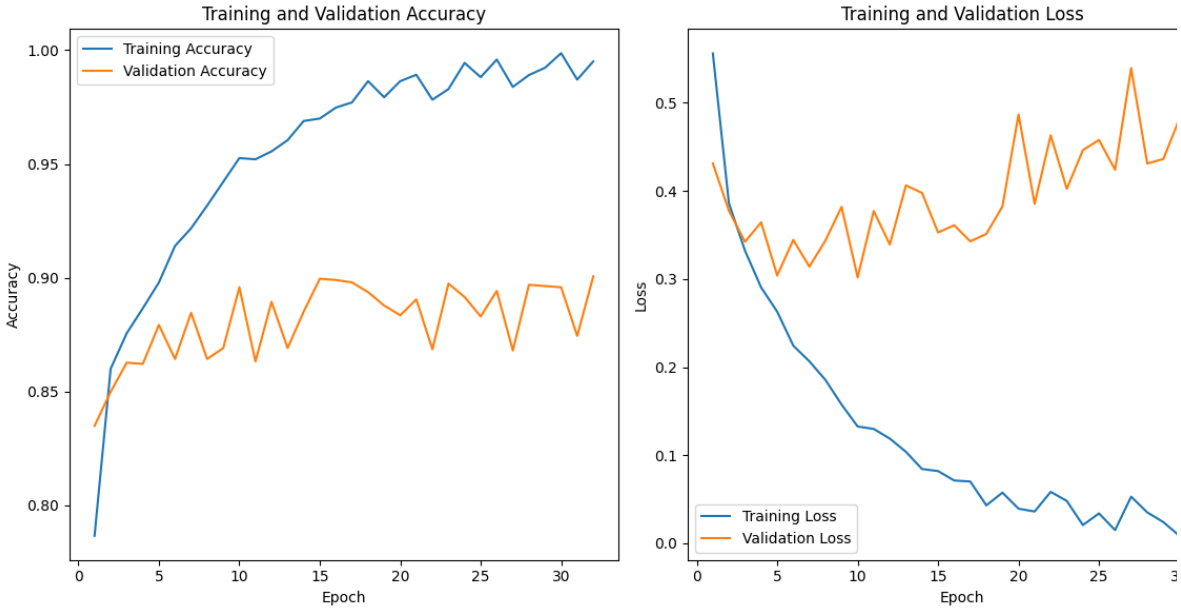

Fig 4.18: Performance Result of mobilenetV2

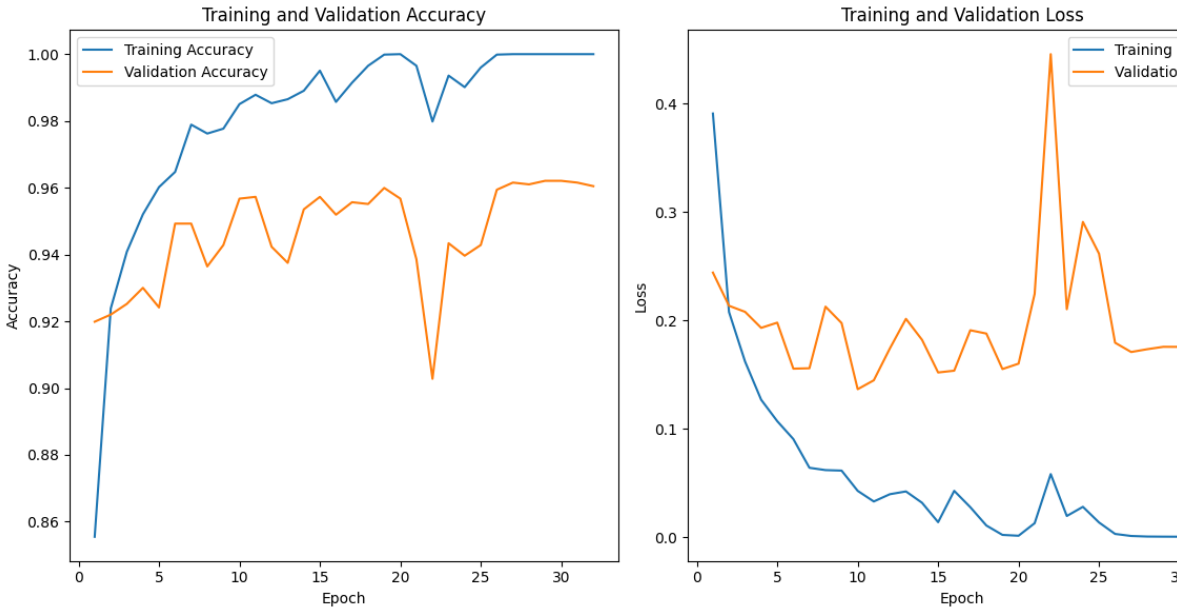

Fig 4.19: Performance Result of Resnet50

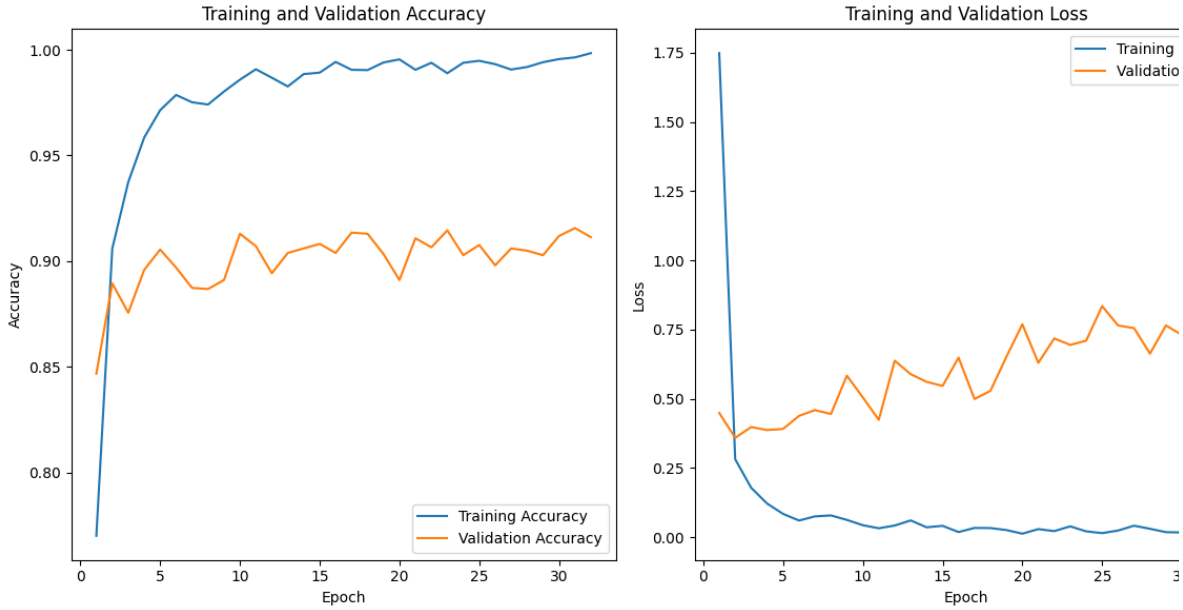

Fig 4.20: Performance Result of VGG19

4.4.2 Performance Metrics

Models	Precision	Recall	F1-Score
Scratched CNN	0.7395	0.7368	0.7378
MobileNetV2	0.9015	0.9022	0.9018
ResNet50	0.9605	0.9605	0.9604
VGG19	0.9078	0.9076	0.9076

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

Models	Precision	Recall	F1-Score
Scratched CNN	0.7403	0.7373	0.7384
MobileNetV2	0.9006	0.9012	0.9008
ResNet50	0.9605	0.9605	0.9604
VGG19	0.9105	0.9103	0.9102

Table 4.15: Precision, Recall, F-1 Score for KNN

Models	Precision	Recall	F1-Score
Scratched CNN	0.7395	0.7368	0.7378
MobileNetV2	0.5823	0.7035	0.6246
ResNet50	0.9599	0.9599	0.9599
VGG19	0.9087	0.9086	0.9085

Table 4.16: Precision, Recall, F-1 Score for AdaBoost

4.5 Result Analysis

Dataset	Best Model	Best Classifier	Best Accuracy
Original Dataset	ResNet50	Random Forest	96.3%
Gaussian Filter Dataset	ResNet50	Random Forest/KNN	97.1%
Laplacian Filter Dataset	MobileNetV2	Random Forest	91.7%
Unsharp mask Filter Dataset	ResNet50	Random Forest/KNN	96%

Table 4.17: Best model and classifier for every dataset

In the Original Dataset, ResNet50 stands out as the best model, achieving impressive accuracies of 96.3% with Random Forest, KNN, and 96.2% with AdaBoost. The Gaussian Filter Dataset showcases ResNet50 again as the top model, achieving remarkable accuracies of 97.1% with Random Forest, KNN, and AdaBoost at 97%. In the Laplacian Filter Dataset, MobileNetV2 emerges as the preferred model with accuracies of 91.7% with Random Forest, KNN, and 91.4% with AdaBoost. Lastly, the Unsharp Mask Filter Dataset demonstrates ResNet50's dominance, reaching accuracies of 96% with Random Forest, KNN, and AdaBoost at 96%. Across classifiers, Random Forest consistently exhibits robust performance, making it a favorable choice for various datasets.

Based on the highest overall accuracy, the Gaussian Filter Dataset with the ResNet50 model and the Random Forest classifier stands out as the most favorable choice. This combination consistently achieves top-notch accuracy across various classifiers, making it a strong candidate for applications prioritizing high predictive performance.

4.6 Visual Representation of prediction

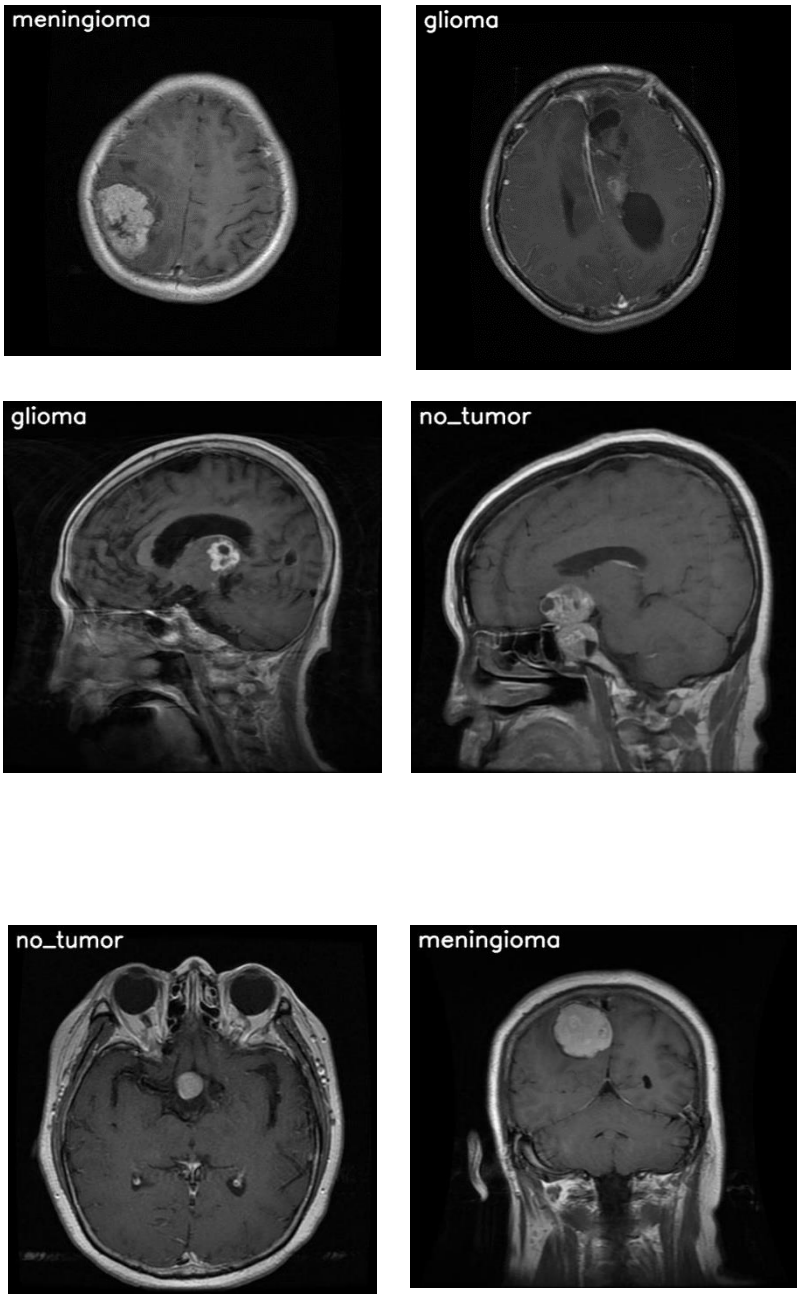

Fig 4.21: visual representation of prediction

Chapter 5

Discussion

This work suggests a novel method for identifying brain cancers from MRI data by combining image processing approaches. By filtering out the noise from the image, the tumor-affected area of the brain can be seen clearly. Classifiers are used in this thesis to classify and label the photos. Random Forest, KNN, and AdaBoost classifiers were employed. Laplacian, Unsharp mask, and Gaussian filters were employed for filtering. For feature extraction, the ResNet50, MobileNetV2, and Scratch CNN models were employed. In all four datasets, Random Forest and Scratched CNN outperform other models and classifiers. The Scratch CNN model with Random Forest classifier on the Laplacian dataset in this study had the maximum accuracy. There is 98.91% accuracy. Unsharp Mask is the second-best dataset, with 98.81% accuracy achieved using Random Forest and Scratch CNN. The Fl score, recall, and precision of several CNN models are also examined in this study.

5.1 Limitations

Taking into account the advancements and promising results reported in this paper, many limitations must be noted:

5.1.1 Limited Dataset Size

The findings' generalizability may be affected by the study's use of a very small dataset. Gaining access to more extensive and diverse datasets may facilitate the testing of the proposed methodology and provide a more comprehensive analysis.

5.1.2 Scope of Tumor Types

It's possible that the study focuses on a certain subset of brain tumor forms, which would limit how broadly it may be applied. Expanding the scope to include other tumor types would increase the effectiveness of the proposed technique.

5.1.3 Limited Clinical Validation

Despite having a high degree of accuracy in identifying brain tumors, the proposed technique may not have gotten enough clinical validation. Comprehensive clinical trials and validation studies with medical professionals and genuine patient data would be necessary to assess how well it functions in real-world scenarios.

5.1.4 Limited Comparison With Existing Methods

Although the accuracy of the proposed method is noteworthy, a direct comparison with other state-of-the-art methods already in use may not be possible. Conducting comparative studies with other established methodologies would provide a benchmark for evaluating the efficacy and productivity of the system being examined.

5.1.5 Hardware and computational Requirements

It's possible that the study does not address the hardware and computing needs needed to put the suggested strategy into practice. Resource or computational limitations may make it more difficult to deploy and implement the system in real time.

Chapter 6

Conclusion

Modern medical science relies heavily on image processing to analyze a wide range of disorders. In this study, MRI pictures are used to detect and diagnose brain cancers. Image processing and machine learning classifiers are used to identify and categorize brain cancers. When it comes to brain tumor identification, the Random Forest classifier and the scratch Convolutional Neural Network (CNN) model perform better than the other methods. In our research, this combination produced the best outcomes.

6.1 Future Work

Our future goal is to improve brain tumor classification accuracy through sophisticated approaches and procedures. We will focus on specific tumor datasets such as meningioma, pituitary adenoma, and craniopharyngioma. We intend to work on other medical imaging, such as CT scans and X-rays. This approach aims to help clinicians make informed judgements about their patients' ultimate treatment. This effort will contribute to implementing a predicted accuracy model into Magnetic. Resonance Imaging (MRI) systems can give patients with information about the model's accuracy along with their results. This study provides insights into the most effective filters for brain tumor MRI pictures. Future study aims to improve the accuracy, effectiveness, and clinical application of the brain tumor detection system, resulting in better patient outcomes and supporting medical professionals with diagnostic decision-making.

Bibliography

[1] Padmavathi, K. and Megala, C., 2015. Detection of brain tumour with filtering techniques. International Journal of Advanced Research in Computer and Communication Engineering, 4(7).
[2] Ravichandran, S. and Rajendran, P., Brain Tumor Research Publication During The Period 2012-2021: A Scientometric Study.
[3] Wu, N., Phang, J., Park, J., Shen, Y., Huang, Z., Zorin, M., Jastrzębski, S., Févry, T., Katsnelson, J., Kim, E. and Wolfson, S., 2019. Deep neural networks improve radiologists' performance in breast cancer screening. IEEE transactions on medical imaging, 39(4), pp.1184-1194.
[4] Lee, E.Q., Selig, W., Meehan, C., Bacha, J., Barone, A., Bloomquist, E., Chang, S.M., De Groot, J.F., Galanis, E., Hassan, I. and Kalidas, C., 2021. Report of National Brain Tumor Society roundtable workshop on innovating brain tumor clinical trials: building on lessons learned from COVID-19 experience. Neuro-oncology, 23(8), pp.1252-1260.
[5]Gamage, Praveen. (2017). Identification of Brain Tumor using Image Processing Techniques - Image Pre-Processing and Segmentation.
6[]Liu, D., Liu, Y. and Dong, L., 2019. G-ResNet: Improved ResNet for brain tumor classification. In Neural Information Processing: 26th International Conference, ICONIP 2019, Sydney, NSW, Australia, December 12–15, 2019, Proceedings, Part I 26 (pp. 535-545). Springer International Publishing.
[7]Belaid, O.N. and Loudini, M., 2020. Classification of brain tumor by combination of pre-trained vgg16 cnn. Journal of Information Technology Management, 12(2), pp.13-25.
[8]Mohan, R., Ganapathy, K. and Rama, A., 2021. Brain tumour classification of magnetic resonance images using a novel CNN based medical image analysis and detection network in comparison with VGG16. Journal of population therapeutics and clinical pharmacology, 28(2).
[9]Rasheed, Z., Ma, Y.K., Ullah, I., Al Shloul, T., Tufail, A.B., Ghadi, Y.Y., Khan, M.Z. and Mohamed, H.G., 2023. Automated Classification of Brain Tumors from Magnetic Resonance Imaging Using Deep Learning. Brain Sciences, 13(4), p.602.
[10]Sharma, A.K., Nandal, A., Dhaka, A., Polat, K., Alwadie, R., Alenezi, F. and Alhudhaif, A., 2023. HOG transformation based feature extraction framework in modified Resnet50 model for brain tumor detection. Biomedical Signal Processing and Control, 84, p.104737.
[11]Shah, M.F.M., 2019. Brain Tumor Detection using Convolutional Neural Network. Ahsanullah University of Science andTechnology.
[12]Methil, A.S., 2021, March. Brain tumor detection using deep learning and image processing. In 2021 international conference on artificial intelligence and smart systems (ICAIS) (pp. 100-108). IEEE.
[13]Ari, A. and Hanbay, D., 2018. Deep learning based brain tumor classification and detection system. Turkish Journal of Electrical Engineering and Computer Sciences, 26(5), pp.2275-2286.
[14]Paul, J.S., Plassard, A.J., Landman, B.A. and Fabbri, D., 2017, March. Deep learning for brain tumor classification. In Medical Imaging 2017: Biomedical Applications in Molecular, Structural, and Functional Imaging (Vol. 10137, pp. 253-268). SPIE.
[15]Rehman, A., Khan, M.A., Saba, T., Mehmood, Z., Tariq, U. and Ayesha, N., 2021. Microscopic brain tumor detection and classification using 3D CNN and feature selection architecture. Microscopy Research and Technique, 84(1), pp.133-149.
[16]Zhao, L. and Jia, K., 2016. Multiscale CNNs for brain tumor segmentation and diagnosis. Computational and mathematical methods in medicine, 2016.
[17]Deshpande, A., Estrela, V.V. and Patavardhan, P., 2021. The DCT-CNN-ResNet50 architecture to classify brain tumors with super-resolution, convolutional neural network, and the

ResNet50.Neuroscience Informatics, 1(4), p.100013.
[18]Ahmad, I.S., Zhang, S., Saminu, S., Wang, L., Isselmou, A.E.K., Cai, Z., Javaid, I., Kamhi, S. and Kulsum, U., 2021. Deep learning based on CNN for emotion recognition using EEG signal.
[19]Khan, H.A., Jue, W., Mushtaq, M. and Mushtaq, M.U., 2021. Brain tumor classification in MRI image using convolutional neural network. Mathematical Biosciences and Engineering.
[20]Çinar, A. and Yildirim, M., 2020. Detection of tumors on brain MRI images using the hybrid convolutional neural network architecture. Medical hypotheses, 139, p.109684.
[21]Huang, Z., Du, X., Chen, L., Li, Y., Liu, M., Chou, Y. and Jin, L., 2020. Convolutional neural network based on complex networks for brain tumor image classification with a modified activation function. IEEE Access, 8, pp.89281-89290.
[22]Bhanothu, Y., Kamalakannan, A. and Rajamanickam, G., 2020, March. Detection and classification of brain tumor in MRI images using deep convolutional network. In 2020 6th international conference on advanced computing and communication systems (ICACCS) (pp. 248-252). IEEE.
[23]Rahmathunneesa, A.P. and Muneer, K.A., 2019, November. Performance analysis of pre-trained deep learning networks for brain tumor categorization. In 2019 9th International Conference on Advances in Computing and Communication (ICACC) (pp. 253-257). IEEE.
[24]Tazin, T., Sarker, S., Gupta, P., Ayaz, F.I., Islam, S., Monirujjaman Khan, M., Bourouis, S., Idris, S.A. and Alshazly, H., 2021. A robust and novel approach for brain tumor classification using convolutional neural network. Computational Intelligence and Neuroscience, 2021.
[25]Kaur, D., Goel, S., Nijhawan, R. and Gupta, S., 2022, February. Analysis of brain tumor using pre-trained CNN models and machine learning techniques. In 2022 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS) (pp. 1-6). IEEE.
[26]Kumari, R. and Srivastava, S.K., 2017. Machine learning: A review on binary classification. International Journal of Computer Applications, 160(7).
[27]Amin, J., Sharif, M., Haldorai, A., Yasmin, M. and Nayak, R.S., 2022. Brain tumor detection and classification using machine learning: a comprehensive survey. Complex & intelligent systems, 8(4), pp.3161-3183.
[28]Saeed, M.U., Ali, G., Bin, W., Almotiri, S.H., AlGhamdi, M.A., Nagra, A.A., Masood, K. and Amin, R.U., 2021. RMU-net: a novel residual mobile U-net model for brain tumor segmentation from MR images. Electronics, 10(16), p.1962.
[29]Rajinikanth, V., Joseph Raj, A.N., Thanaraj, K.P. and Naik, G.R., 2020. A customized VGG19 network with concatenation of deep and handcrafted features for brain tumor detection. Applied Sciences, 10(10), p.3429.
[30]Zhang, L., Zhang, H., Rekik, I., Gao, Y., Wang, Q. and Shen, D., 2018. Malignant brain tumor classification using the random forest method. In Structural, Syntactic, and Statistical Pattern Recognition: Joint IAPR International Workshop, S+ SSPR 2018, Beijing, China, August 17–19, 2018, Proceedings 9 (pp. 14-21). Springer International Publishing.
[31]Faraz, N., Naz, B. and Memon, S., 2022. Data mining approach for detection and classification of brain tumor. Mehran University Research Journal Of Engineering & Technology, 41(1), pp.53-64.
[32]Khagi, B. and Kwon, G.R., 2021. Convolutional neural network-based natural image and MRI classification using Gaussian activated parametric (GAP) layer. IEEE Access, 9, pp.96930-96947.
[33]Rajinikanth, V., Satapathy, S.C., Fernandes, S.L. and Nachiappan, S., 2017. Entropy based segmentation of tumor from brain MR images–a study with teaching learning based optimization. Pattern Recognition Letters, 94, pp.87-95.
[34]Mohan, R., Ganapathy, K. and Rama, A., 2021. Brain tumour classification of magnetic resonance images using a novel CNN based medical image analysis and detection network in comparison with VGG16. Journal of population therapeutics and clinical pharmacology, 28(2).
[35]Arshed, M.A., Shahzad, A., Arshad, K., Karim, D., Mumtaz, S. and Tanveer, M., 2022. Multiclass Brain Tumor Classification from MRI Images using Pre-Trained CNN Model.
[36]Lathashree, P.S., Puthran, P., Yashaswini, B.N. and Patil, N., 2022. Brain MR Image Segmentation for Tumor Identification Using Hybrid of FCM Clustering and ResNet. In Inventive Systems and

Control: Proceedings of ICISC 2022 (pp. 231-245). Singapore: Springer Nature Singapore.
[37]Saeed, M.U., Ali, G., Bin, W., Almotiri, S.H., AlGhamdi, M.A., Nagra, A.A., Masood, K. and Amin, R.U., 2021. RMU-net: a novel residual mobile U-net model for brain tumor segmentation from MR images. Electronics, 10(16), p.1962.
[38]Arshed, M.A., Shahzad, A., Arshad, K., Karim, D., Mumtaz, S. and Tanveer, M., 2022. Multiclass Brain Tumor Classification from MRI Images using Pre-Trained CNN Model.
[39]Rajinikanth, V., Joseph Raj, A.N., Thanaraj, K.P. and Naik, G.R., 2020. A customized VGG19 network with concatenation of deep and handcrafted features for brain tumor detection. Applied Sciences, 10(10), p.3429.
[40]Gayathri Shrikanth & Sanika Mhadgut. Brain Tumor Detection using FastAl and OpenCV. May 5,2020. Brain Tumor Detection using FastAI and OpenCV
[41]Pravitasari, A.A., Iriawan, N., Almuhayar, M., Azmi, T., Irhamah, I., Fithriasari, K., Purnami, S.W. and Ferriastuti, W., 2020. UNet-VGG16 with transfer learning for MRI-based brain tumor segmentation. TELKOMNIKA (Telecommunication Computing Electronics and Control), 18(3), pp.1310-1318.
[42]Deshpande, A., Estrela, V.V. and Patavardhan, P., 2021. The DCT-CNN-ResNet50 architecture to classify brain tumors with super-resolution, convolutional neural network, and the ResNet50. Neuroscience Informatics, 1(4), p.100013.
[43]Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M. and Kudlur, M., 2016. {TensorFlow}: a system for {Large-Scale} machine learning. In 12th USENIX symposium on operating systems design and implementation (OSDI 16) (pp. 265-283).
[44]Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V. and Vanderplas, J., 2012. Scikit-learn: Machine learning in python. ArXiv. Org.
[45]Culjak, I., Abram, D., Pribanic, T., Dzapo, H. and Cifrek, M., 2012, May. A brief introduction to OpenCV. In 2012 proceedings of the 35th international convention MIPRO (pp. 1725-1730). IEEE.