FEATURE EXTRACTION OF TUMOR IMAGE BASED ON CONVOLUTIONAL NEURAL NETWORK

ABSTRACT

Healthcare is a high priority sector where people expect the best care and services regardless of cost. Medical images are important in helping doctors to make timely decisions and right diagnosis to treat patients effectively. Deep learning is now providing exciting medical imaging solutions with great accuracy, after it's success in other real-world applications. Intelligent algorithms allow to quickly detect lesions in medical imaging, and it is significant to extract features from the images. A vast amount of data is analyzed to obtain processing results for medical image feature extraction, that aids doctors in making highly accurate case diagnosis. In this project, brain scanned MRI image is considered as an input to the system. The basic architecture of convolutional neural network (CNN) is constructed by focusing on image feature extraction using a CNN. The image remains fixed with reference to the coordinate system as it shifts and rotates. The method can accurately explain the texture features of the tumor images's shallow layer, thereby enhancing the robustness of the image region description. CNN will classify the image for benign or malignant. The CNN architecture exhibited satisfying accuracy on the dataset. In order to obtain maximum accuracy, Gabor filter algorithm is used in CNN. It can be seen from the experimental results that the CNN along with Gabor filter outperformed in tumor image feature extraction.

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

The major and significant carrier of modern medical alternative language description is the medical image. With extremely intuitive expression and objective description, it gradually plays an essential part in the medical area. The term “tumor” refers to abnormal tissue growth that can be benign (non-tumorous) which does not spread to other parts of the brain or malignant (tumorous) which spreads rapidly to other regions of the body easily and leads to immediate death. Detecting the tumors at an early stage plays a vital role in the field of medical industry and helps doctors in saving lives. The advancement in the field of technology and artificial intelligence provides a wide scope by integrating different methodologies and algorithms in image processing. Artificial intelligence has gained a lot of attention in recent years as a result of its achievements in the field of intelligent medicine. Artificial intelligence-based classification of MR images has become a prominent area in medical image processing research. In an image, texture is fundamental and valuable information element and is a key condition for characterizing the visual content. The features of the image are extracted and fed into the CNN model, trained, and validated and finally classified as benign or malignant tumor.

1.1 Problem

Medical imaging is a rapidly evolving area of engineering. Magnetic Resonance Imaging (MRI) is one of the most dependable imaging techniques for medical diagnosis. The procedures for detecting brain tumors are feature extraction and classification. Traditional manual feature extraction, such as intensity and texture aspects of brain tumor images, were frequently used in prior research. Traditional feature extraction techniques, necessitate expert knowledge and experience in specialized disciplines. The efficiency of the system will be reduced if features are extracted manually. It was critical to develop an automatic and effective brain tumor MR image classification approach to aid physicians in their decision-making. The disadvantage is overcome by deep learning techniques.

Deep learning-based feature extraction algorithms have shown to be effective in real-world medical image processing applications. Later, a CNN method was created for the autonomous extraction of the most common tumor types but without preprocessing. Many researchers developed a high-performance object detector based on CNN that produced 70% improvement in accuracy over prior research. However, achieving maximum accuracy and efficiency remained the major challenge due to the complexity and variance of MR images.

1.2 Project Statement

The main objective of this project is to extract features of tumor MRI image of a brain that contains tumor or non-tumor by incorporating convolutional neural network technique to achieve the classification along with Gabor filter to enhance high accuracy. The proposed system employs a learning method to extract features from a given dataset, as well as a saved model to identify tumor classification details and tumor detection.

1.3 Approach

Dataset which contains 253 MR Images of brain includes 155 images of tumorous and 98 images of non-tumorous is considered for implementation. Data Preprocessing is done in order to convert raw data into understandable format which can be fed into the neural network.

Dataset is split into 80% train set which can be used for training the model and 20% test set which validates the model. Features are extracted from the images so that performance can be improved. CNN model is built, trained with the images, and validated. The accuracy seems to be not upto the expectation. Introduced Gabor filter algorithm and appended to CNN model, trained with images, and validated. The confidence score is measured and analyzed against the observed values.

1.4 Organization of this Project Report

The organization of this project report as follows:

- Chapter 2 covers the background, keyconcepts, related work and literature
- Chapter 3 provides the architecture, high-level design, flow chart implementation.
- Chapter 4 includes the methodology, results, and analysis
- Chapter 5 concludes and talks about future work.

2. BACKGROUND

2.1 Key Concepts

For this project, the following important concepts were employed for the study and exploration of machine learning concepts.

2.1.1 Convolutional Neural Network

A Convolutional Neural Network (CNN) is a Deep Learning algorithm that was designed primarily for processing with images and videos. It takes images as inputs, extracts**,** and learns the image's features, and then classify them using the learned features. Multiple building blocks, such as convolution layers, pooling layers, and fully connected layers, constitute a convolutional neural network.

The computer sees an array of pixels instead of an image. If the image input size is 256X256X1, where 256 represents width, 256 represents height, and 1 represents single channel value. Each of these numbers is given a value between 0 and 255 by the computer. The intensity of the pixel at each place is described by this value.

- Convolutional layer: The convolutional layer is the most important component of a CNN because it is where the majority of the computation takes place. It requires input data, a filter, and a feature map, among other things. The image is converted into numerical data by the convolutional layer, which allows the neural network to analyze and extract useful patterns.

- Pooling layer: Pooling layer is a dimensionality reduction technique that reduces the number of parameters in the input. The pooling process passes a filter across the entire input, similar to the convolutional layer, however this filter does not have any weights. Instead, the kernel uses an aggregation function to populate the output array with the values from the receptive field.
- Fully Connected layer: Fully Connected Neural Networks are composed of layers of nodes, each of which is connected to the previous layer's nodes. This network is appended to CNN architecture to make a prediction based on convolved learned features.

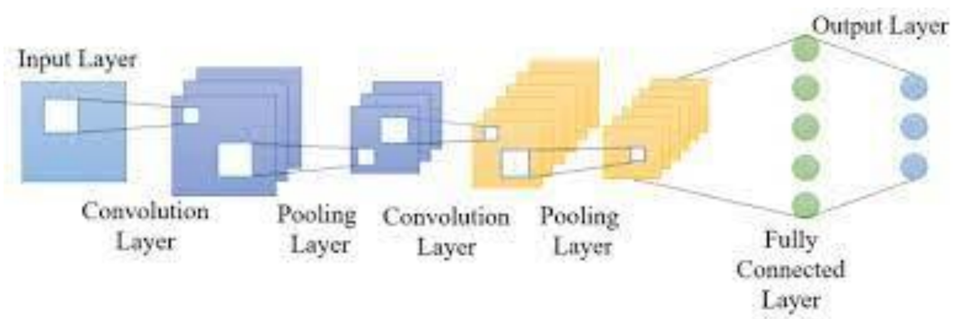


Figure 1: CNN Architecture

2.1.2 Image Feature Extraction Based On CNN

The parts or patterns of an object in an image that help to identify it are called features. CNN is a neural network that extracts input image features and classifies them using another neural network. The feature extraction network uses the input image. The neural network uses the extracted feature values to classify the data. The feature is obtained via CNN using a convolution kernel, whose parameters are learned. As a result, it has a high ability to express features. In this project, texture features are extracted to feed the neural network.

2.1.3 Gabor Filter

A Gabor filter (in image processing), named after Dennis Gabor, is a linear texture analysis filter that examines if the image contains any specific frequency content in specific directions in a confined region around the point or region of research.

Many modern vision scientists argue that Gabor filters' frequency and orientation representations are identical to those of the human visual system, despite the lack of scientific data and a functional explanation to support this claim.

A 2D Gabor filter is a Gaussian kernel function modulated by a sinusoidal plane wave in the spatial domain.According to some scientists, Gabor functions can be used to mimic basic cells in the visual cortex of mammalian brains. As a result, some believe that picture processing with Gabor filters is similar to perception in the human visual system.

Applications of 2D Gabor filters in image processing:

- Gabor features are useful for recognizing the script of a word in a multilingual document when it relates to document image processing.
- It has also been used to recognize facial expressions.
- In image processing applications such as optical character recognition, iris recognition, and fingerprint recognition, the Gabor space proves useful.
- Between objects in an image, the relationships between activations for a certain spatial region are quite apparent.

- In order to generate a sparse object representation, important activations can be recovered from the Gabor space.

2.2 Related Work or Literature Review

2.2.1 Brain Tumor Classification Employing Modulated Gabor Filter Banks

From the paper[1] the authors have combined Gabor filter orientation and scale dynamics, created a novel deep learning architecture for MR brain tumor classification. Due to the incorporation of steerable Gabor filter dynamics, the Gabor-modulated convolutional filter network is robust to orientation and scale variability. This allows their architecture to learn smaller feature maps, reducing the number of network parameters required. They examined the performances of the proposed model using a number of performance metrics, including accuracy, F1 score, sensitivity, and specificity. The experimental results support their assertion that the suggested technique improves current state-of-the-art procedures.

2.2.2 Brain Tumor Image Classification Using CNN and SVM

From paper [2] the authors proposed methods that employs a convolutional neural network (CNN) and a support vector machine (SVM) to classify brain tumor images into three subtypes: meningioma, glioma, and pituitary. The extraction is done by CNN utilizing convolution layers, and the amount of feature increases as the depth increases. In contrast, features in SVM are retrieved based on the sort of texture or pattern in the image, and classes with similar features can be easily classified. So, using the training set, CNN and SVM-based architectures are chosen to train, and then the trained model is validated using the validation and testing datasets. Finally, the classifier models' accuracy is calculated. As a result of its much-improved performance on many measures such as ROC-AUC, Precision, Recall, and Accuracy, they concluded that CNN is the best alternative for the most exact and trustworthy classification.

3. ARCHITECTURE

3.1 High Level Design

The architecture which would be utilized to construct a system is described in high-level design. The architecture diagram depicts the overall system, defining the main components and their interfaces that will be designed for the product. High-level design is defined in the context of the development environment in which it will be implemented.

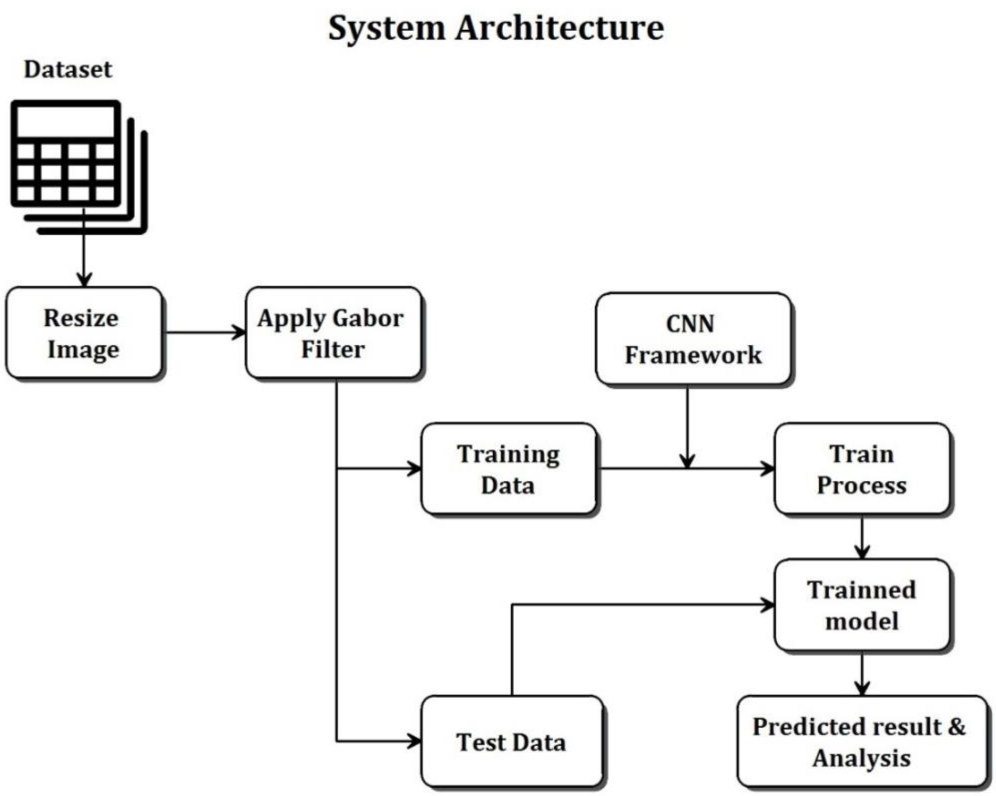


Figure 2: System Architecture

Dataset is a collection of data. Data preprocessing is the process of converting raw data into a suitable form. As we cannot work with raw data, it is important step in data mining. Before using learning algorithms, ensure the data is of great quality. Feature engineering is the process of choosing, modifying, and transforming raw data into features that may be used in learning procedures. It may be necessary to build and train features in order for machine learning (ML) to perform well. A strategy for measuring the performance of a ML algorithm is the train-test split. It can be used for any supervised learning technique and can be utilized for classification problems. The technique involves taking a dataset and dividing into two sets. A training model is dataset used to train a ML algorithm. A fully trained model is evaluated on a testing dataset. The outcome is predicted by using the test data on the trained model.

3.2 Flow Chart Diagram

A flowchart is a diagram that depicts the steps, sequences and decisions that occur during the execution of a process or workflow. Also called as process maps, this type of tool depicts one or more inputs and changes them into outputs through a number of processes with branching options.

The dataset is the input to the system. Apply the methodology or the process to the preprocessed data. Features are extracted from the given dataset. Dividing the dataset into train-test set. Train the CNN model and after fully trained, input the test set in order to predict the outcome.

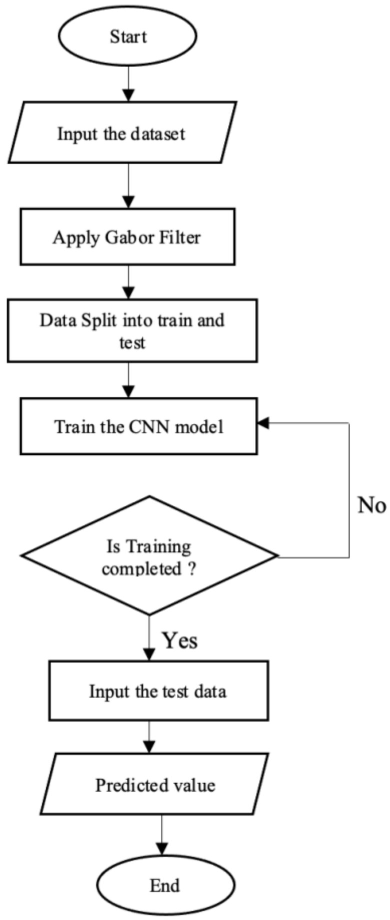


Figure 3: Flow Chart Diagram

3.3 Implementation

Modules

- Data Gathering and Preprocessing
- Feature Selection and Data Preparation
- Model Construction and Model Training
- Model Validation and Result Analysis
- Model Saving

Data Gathering and Preprocessing

The Feature Extraction of Tumor Image Based on CNN project is designed based on the dataset of Brain MRI Images for Brain Tumor Detection provided by Kaggle. There are 253 images as the input data samples and consists of the type .jpg, .jpeg or .png. The dataset has two labels: yes and no where yes contains all the tumorous 155 images and no contains all the non-tumorous 98 images. In this project, 2D convolutional Neural Network is considered in order to classify 2D MRI brain scan images as tumorous(malignant) or non-tumorous(benign). The input to the model is of 256X256X1 single-channel pixel arrays. The primary data gathered from the online sources and remains in raw format. Raw data contains inaccurate or missing data, presence of noisy data and even inconsistent information.

https://www.kaggle.com/saumya5679/brain-tumor-mri-inceptionresnetv2-85-accuracy/data

Import the libraries:

i. numpy as np - In python, Numpy package is the foundation for scientific computing. It includes a multidimensional array object with high performance as well as utilities for interacting with these arrays.

ii. matplotlib.pyplot as plt - In python, the matplotlib is a graphical plotting library and pyplot is a module with set of functions that allow matplotlib to behave similar to MATLAB. Every pyplot function modifies a figure in some way like creating a figure, plotting area in figure, charting certain lines in a plot area and so on. plt is an alias used.

iii. %matplotlib inline - Matplotlib's backend is set to inline by %matplotlib inline, backend: The output of plotting commands is presented inline within frontends like the jupyter notebook, just under the code cell that generated it, using this backend.

iv. seaborn as sns - Seaborn is Matplotlib-based python data visualization package. The alias sns is referred instead of seaborn.

v. cv2 as cv - opencv-python, is the module import name for cv2. Traditional OpenCV contains a lot of difficult procedures that need you to develop the module from scratch, which isn't essential. opencv-python library is preferred.

vi. tensorflow.keras.utils - Converts a class vector (integers) to binary class matrix.

vii. tensorflow.keras.layers - Types of convolution layers

viii. tensorflow.keras.models - Keras model architectures

ix. tensorflow.keras.optimizers - Changing the attributes like weights & learning rate to reduce the losses

x. sklearn.model_selection - Provides train/test indices to split data into train/test sets.

xi. sklearn.metrics - Compute the metrics like confusion_matrix, accuracy

Data Preprocessing is a method for converting raw data into a clean data set. Whenever the data is collected from various sources, it will be in raw format which is not feasible for the analysis. Data Preprocessing consists of:

i. Loading the image dataset - learn the images and label information from the dataset and extract the path and categories.

ii. Data Normalization and Reshaping - images consists of matrices of pixel values. Grey images are single matrix of pixels and has a range from 0 to 255. Larger values can slow the training process and it's good to normalize the pixel values so that it will be value of 0 to 1. This is achieved by dividing all pixels by 255. The reshape( ) method changes the shape of an array.

Feature Selection and Data Preparation

Feature engineering is the process of creating features for machine learning algorithms utilizing domain knowledge of the data. If done correctly, feature engineering improves the prediction potential of machine learning algorithms by generating features from data that aid the machine learning process. When constructing a predictive model, feature selection reduces the number of input variables so that it can enhance the performance of the model and also lower the computational cost.

The classification of data is the process of arranging data into groups and classes depending on the specific characteristics. It might be based on numerical characteristics or attributes. So, in order to determine whether the training data contains the correct label, also known as a target, we must visualize the prepared data.

Next, we will slice a single data set into a train set and test set.

- **Train set -** a subset to train a model.
- **Test set -** a subset to evaluate the trained model.

Make sure that test set meets the following two conditions:

- ✓ Is it large enough to produce statistically significant results?
- ✓ Is it representational of the entire data set? or don't choose a test set that differs from the train set.

The dataset is partitioned into train and test sets in order to use them in a keras model. Splitting the input dataset into two different sections: 80% - train set and 20% - test set. The train dataset consists of yes (tumorous) with 132 images and no (non-tumorous) with 77 images whereas test dataset consists of yes (tumorous) with 23 images and no (non-tumorous) with 21 images.

Model Construction and Model Training

A machine learning model is constructed by learning and generalizing from training data, then applying that knowledge to new data which has never seen before in order to generate predictions and accomplish the goals. The meticulously prepared data is used for model building. Build a CNN model with below configuration.

Model Architecture:

Input image size: 256X256X1 – gray image, single channel

The size of the kernels in Conv2D layers is (5, 5), padding =”same”

pool_size of (2, 2) in all MaxPooling2D layers.

Model: ”sequential”

Layer type	Output Shape	Number of Parameters
Conv2D	(None, 256, 256, 64)	1664
MaxPooling2D	(None, 128, 128, 64)	0
Conv2D	(None, 128, 128, 32)	51232
MaxPooling2D	(None, 64, 64, 32)	0
Conv2D	(None, 60, 60, 16)	12816
Dropout	(None, 60, 60, 16)	0
Flatten	(None, 57600)	0
Dense	(None, 1024)	58983424
Dropout	(None, 1024)	0
Dense	(None, 2)	2050

As it is a binary classification, used "relu" activation function on the hidden layers and the "softmax" function on the output layer. And there are two output labels, the Adam optimizer is utilized, which is a built-in keras optimizer for stochastic gradient descent. Categorical crossentropy is used as a loss function.

The process of training an ML model involves providing training data for the algorithm to learn from. The model artifact developed by the training process is referred to as an ML model. The correct output, also known as a target, must be included in the training data. The learning algorithm searches the training data for patterns that connect the input data to the target, and then generates an ML model that captures these patterns.

Model Validation and Result Analysis

The model is validated on a new set of data during the testing phase. There are two separate datasets for training and testing. Building a machine learning model with the purpose of having it perform effectively is the goal. On the training set, as well as on new data in the test set, the model should perform well. Real-time data for prediction is passed once the construction model has been evaluated. Once prediction is done, then output should be analyzed to find out the critical information.

Every epoch, the model is trained with an image dataset for a defined number of epochs, with accuracy and loss of the training as well as validation. The accuracy and loss graphs are plotted. The trained model is evaluated against the test dataset and to find the tumor classification details.

Generate the classification report by computing the accuracy, precision, f1score, recall and confusion_matrix.

Model Saving

Tumor detection model structure is saved as a json file and also the weights as additional file. Later while evaluating, the model is retracted back in order to test the user's input.

From this CNN architecture, the accuracy is only 80% and not as expected. In order to improve the accuracy for the trained model, Gabor filter algorithm is used in CNN.

Gabor Filter

An approach for improving image edge detection is the Gabor Filter in edge detection. To achieve more accurate results, Gabor filter was utilized. Consider Gabor Filter as a sine wave overlaid over a Gaussian bell curve in 2D space. The sine wave is directed in 2D space. The bell curve, on the other hand, peaks at a place and then curves out in all directions. When you combine the two, you get a directed wave that focuses on a specific point in space within a given area. When applied this to an image, there are number of beneficial properties that were discovered:

- It enhances pixels on the edges.
- Pixels in close proximity are suppressed.
- It is directional.
- Pixels that are further away have less impact (Local area).

Gabor filter applies each filter to the pixels of an image.

Initialising the parameters of gabor filter:

```
gabor_1 = cv.getGaborKernel((18, 18), 1.5, np.pi/4, 5.0, 1.5, 0, ktype=cv.CV_32F)
```

Applying gabor filter:

```
filtered_img_1 = cv.filter2D(img, cv.CV_8UC3, gabor_1)
```

Using Gabor Filters in edge detection, resulted in significant improvements. Lines are often cleaner, and object edges are more easily detected.

- **<u>Visualization without Gabor filter</u>**

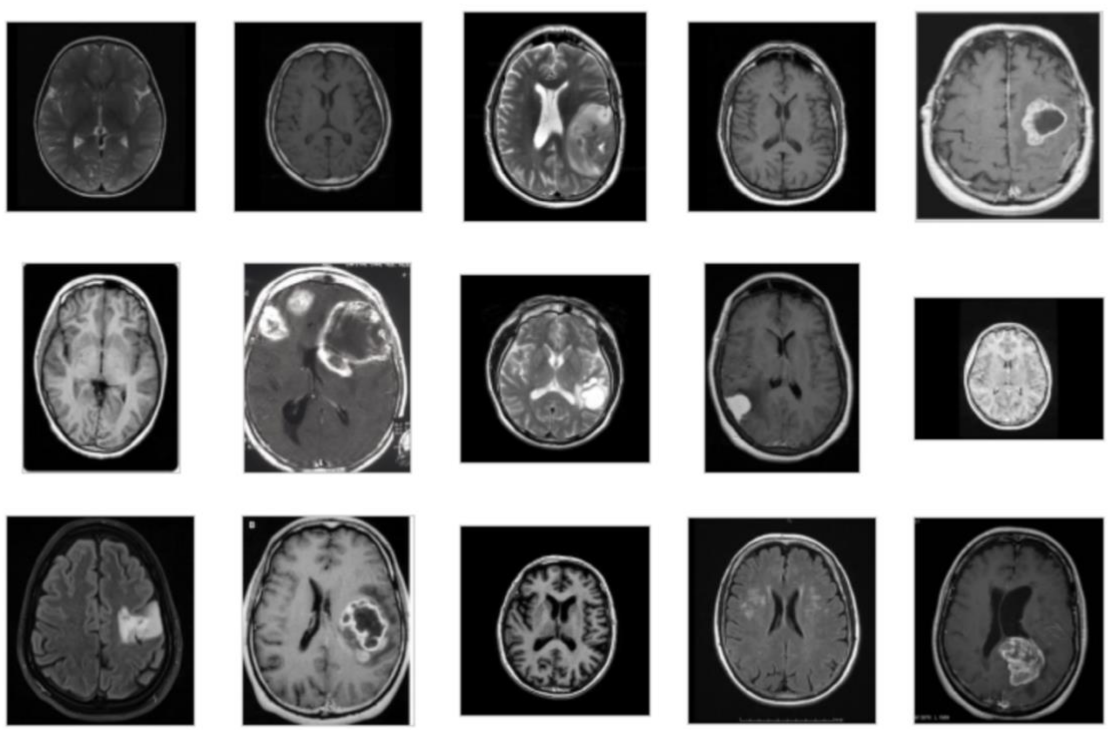


Figure 4: Visualization without Gabor filter

➢ Visualization with Gabor filter

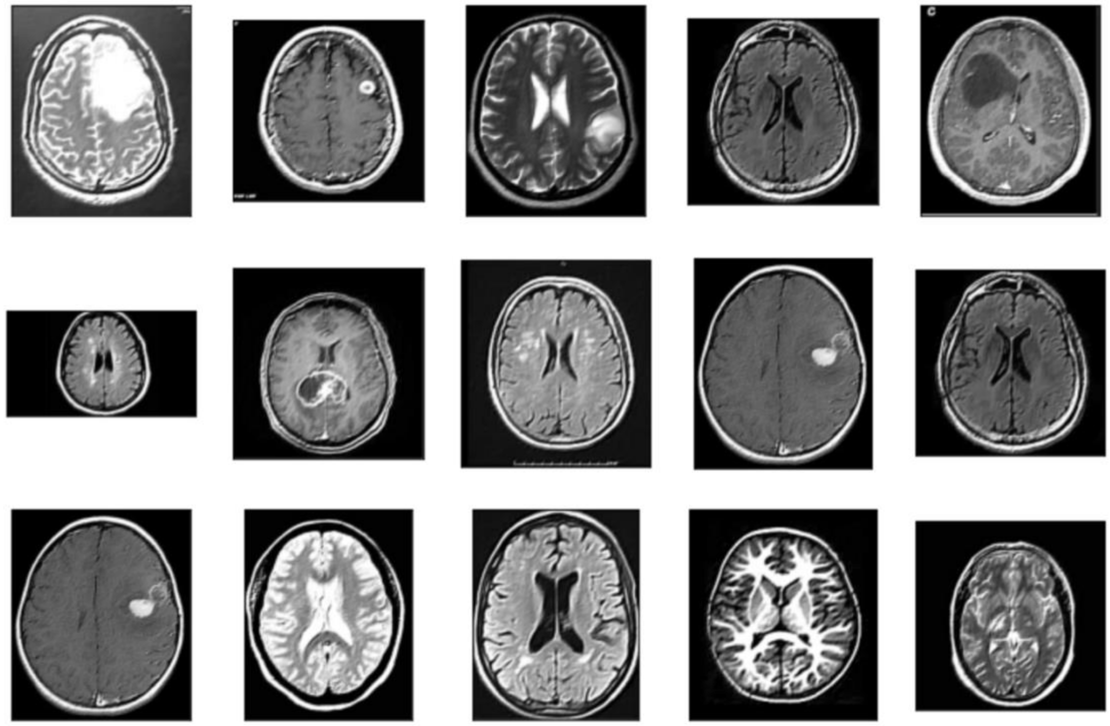


Figure 5: Visualization with Gabor filter

4. METHODOLOGY, RESULTS AND ANALYSIS

In this phase, test and measure the performance of the classification model on the prepared dataset. Accuracy is used to measure the effectiveness of classifiers in order to evaluate the performance of built classification and compare it to the current approaches. Evaluating the power of model prediction on a new instance after model building is a critical challenge. One might try numerous model types for the same prediction problem and then compare the prediction performance to determine which model is the best to utilize in a real-world decision-making circumstance (e.g., accuracy).

Accuracy, recall, and other performance indicators are widely used to evaluate a predictor's performance. The most often used performance measures will be described first, followed by an explanation and comparison of certain well-known estimating approaches. Predictive Modeling Performance Metrics, a confusion matrix is the major source of performance metrics in classification problems. A confusion matrix for a two-class classification problem is shown in the diagram above.

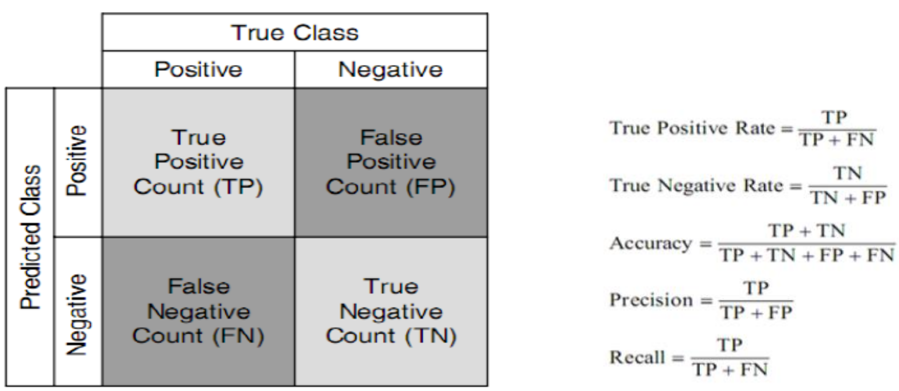


Figure 6: Confusion Matrix and Formulae

The numbers along the diagonal from upper-left to lower-right represent correct decisions, whereas the numbers outside this diagonal represent errors, as seen in the diagram above.

Recall

A true positive rate or recall is calculated by dividing the number of correctly classified positives (the true positive count) by the total number of true positives and false negatives.

$$Recall = \frac{TP}{TP + FN}$$

Precision

The number of correct positive predictions divided by the total number of positive predictions produces precision.

$$Precision = \frac{TP}{TP + FP}$$

F1 Score

It employs harmonic mean of precision and recall. It considers both false positives and negatives.

$$F1\ score = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}} = \frac{2 * (Precision * Recall)}{(Precision + Recall)}$$

Accuracy

The total number of two correct predictions (TP + TN) divided by the total number of datasets represents accuracy.

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

4.1 Methodology

Testing Environment Specifications

Operating System	:	macOS Monterey
Processor	:	Intel Core i5
RAM	:	8GB
Coding Language	:	Python
Tools	:	Homebrew
IDE	:	Jupyter Notebook

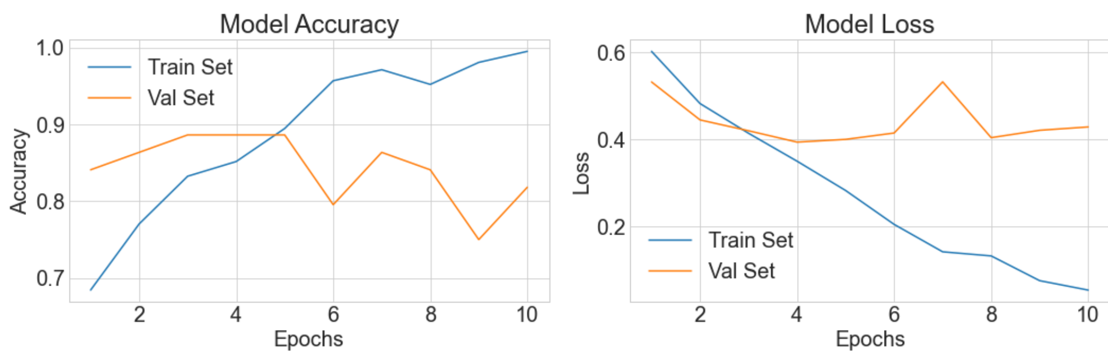
List of Test cases

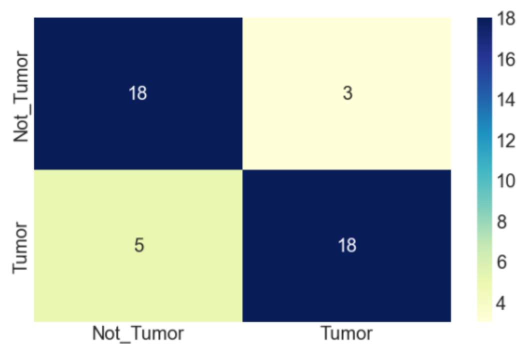
Id	**Test Case Title**	**Test Input**	**Result**	**Remarks**
1	Data Upload and Dataset File Path	Image files uploaded	Success	Pass
2	Data Cleaning	Raw Dataset	Cleaned Data	Pass
3	Data Preparation and Splitting	Dataset and Split Train/Test Ratio	Train and Test dataset created successfully	Pass
4	Model Construction and Training	Training Algorithm	Model trained successfully using Train dataset	Pass

5	Model Validation	Trained Model	Model validated against Test dataset	Pass
6	Display Result	Model Performance Statistics	Classification accuracy and error rate with plot	Pass

4.2 Results

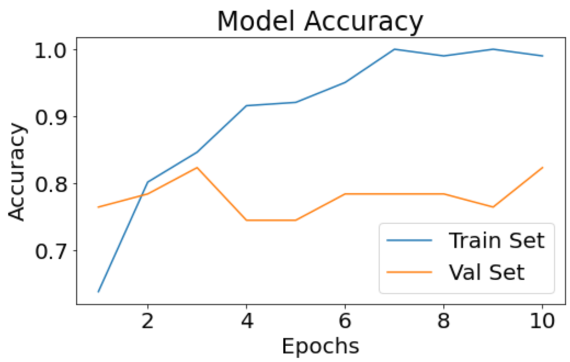
<u>CNN model without Gabor Filter</u>

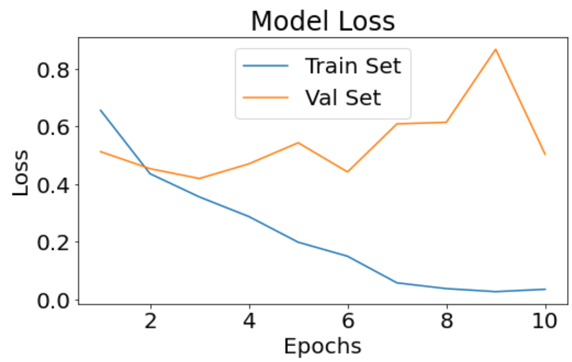


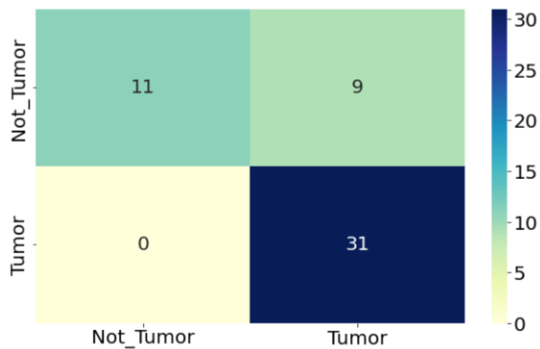


Accuracy is not as expected from the CNN prediction model. So experimented with the neural network design i.e., implementing the Gabor filter algorithm to the CNN prediction model in order to check how a higher efficiency can be achieved.

CNN model with Gabor Filter







4.3 Analysis

User has to input either tumor(yes) or not tumor(no) image to the CNN model with Gabor filter. The Gabor CNN algorithm can only predict “Not_Tumor” or “Tumor” with confidence score. A Confidence Score is a number between 0 and 1 that indicates the probability that a Machine Learning model's output is valid and therefore will meet a user's request.

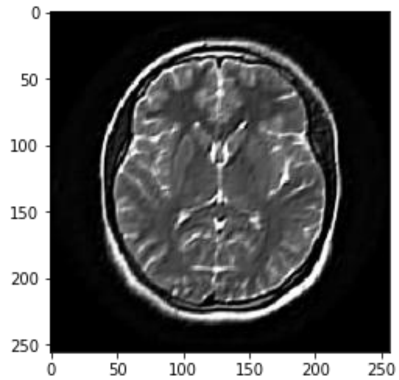


Figure 7: User Input Image

When user feeds the above image to the Gabor model, then it predicts the class as Not_Tumor with confidence score 1.0 and this is analyzed against observed value which turns out to be true. Hence, the CNN model with Gabor filter predicts correctly.

5. CONCLUSION

5.1 Summary

Convolutional neural networks are still a subject of growing interest of automated tumor detection. It is essential for radiologists to acquire a solid knowledge of convolutional neural networks in order to be prepared to use these technologies in medical practice in the future. As proposed in this project, investigated various deep learning technologies, and combined them in order to improve the detection of brain tumors. The CNN combined with the Gabor filter enhanced the accuracy in tumor image feature extraction, as demonstrated by the experimental data. Quantitative results on the brain tumor dataset, which yields confidence score of 1.0, demonstrate the efficiency of this approach. After classification and feature extraction, model validation proved that the findings were accurate.

5.2 Contributions -or- Potential Impact

- The most typical medical application of brain tumor prediction is quantitative image analysis.
- The CNN with Gabor filter being more precise and robust model predict medical outcomes including survival and treatment response.
- In manual detection of tumor, one needs high level of expertise in tumor analysis, whereas the Gabor CNN model automates the detection of brain tumor using MRI’s.

5.3 Future Work

In this project, the model performed well on smaller dataset. in the future, would like to validate the model with a larger dataset. The algorithm may be modified to work with 3-dimensional brain images allowing for more effective image segmentation and tumor image stage identification. This could be used to other fields of radiology by further developing segmentation techniques in tumors.

REFERENCES

[1] Singh, R., Goel, A. & Raghuvanshi, D.K. Computer-aided diagnostic network for brain tumor classification employing modulated Gabor filter banks. *Vis Comput* **37,** 2157–2171 (2021). https://doi-org.aurarialibrary.idm.oclc.org/10.1007/s00371-020-01977-4

[2] S. K. Baranwal, K. Jaiswal, K. Vaibhav, A. Kumar, and R. Srikantaswamy, "Performance analysis of Brain Tumour Image Classification using CNN and SVM," *2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)*, 2020, pp. 537-542, doi: 10.1109/ICIRCA48905.2020.9183023.

[3] A. Yang, X. Yang, W. Wu, H. Liu and Y. Zhuansun, "Research on Feature Extraction of Tumor Image Based on Convolutional Neural Network," in IEEE Access, vol. 7, pp. 24204-24213, 2019, doi: 10.1109/ACCESS.2019.2897131.

[4] M.R.Avendi,A.Kheradvar,andH.Jafarkhani,‘‘Automaticsegmentation of the right ventricle from cardiac MRI using a learning-based approach,’’ Magn. Reson. Med., vol. 78, no. 6, pp. 2439–2448, Dec. 2017.

[5] A. Rafay, M. Suleman, and A. Alim, "Robust review rating prediction algorithm based on machine and deep learning: Yelp dataset," in Intern. Conf. on Emerging Trends in Smart Technologies, March 26-27, 2020, pp. 8138-8143.

[6] https://www.ibm.com/cloud/learn/convolutional-neural-networks

[7] https://www.freedomvc.com/index.php/2021/10/16/gabor-filter-in-edge-detection/