Brain Tumor Detection using Convolutional Neural Network (CNN)

A Project Report

Submitted in Partial Fulfillment of the Requirements for ETE 3200 of

Bachelor of Science (B.Sc.)

in

Electronics & Telecommunication Engineering (ETE)

Submitted by:

Name: **Tanjib Ahmed**

Roll: **1804023**

Supervisor:

Md.Rakib Hossain

Assistant Professor,

Dept. of ETE, RUET

Dept. oF Electronics & Telecommunication Engineering (ETE)

Rajshahi University of Engineering & Technology (RUET)

Kazla, Rajshahi-6204

March 2023

Dept. of Electronics & Telecommunication Engineering (ETE)
Rajshahi University of Engineering & Technology (RUET)
Kazla, Rajshahi-6204

CERTIFICATE

I hereby certify that the Project titled "Brain Tumor Detection using Convolutional Neural Network (CNN)" which is submitted by Tanjib Ahmed, **Roll: 1804023** for the fulfillment of the requirements for awarding of the degree of Bachelor of Science (B.Sc.) is a record of the project work carried out by the students under my guidance & supervision.

Place : Rajshahi,
Bangladesh
Date : March 2023

_____

Md.Rakib Hossain
(Supervisor)
Assistant Professor
Department of ETE
Rajshahi University of Engineering & Technology (RUET)

ABSTRACT

A brain tumor is one of these tumors. It has an impact on the brain, neurological system, glands, and brain-surrounding membranes. Imaging and pathology can both be used to diagnose tumors. Magnetic resonance imaging (MRI), which produces cross-sectional images of the brain, is used to image brain tumors. As a result of the great degree of visual diversity and similarity between brain tumors and normal tissues, it is difficult to distinguish tumor regions from surrounding normal tissue when extracting them from photographs. In this study, we built a model based on artificial convolutional neural networks that collect magnetic resonance pictures and analyze them using matrices operations and mathematical formulas. This neural network estimates the likelihood of a brain tumor by using a real-time dataset with diverse tumor sizes, locations, shapes, and different image intensities which include magnetic resonance pictures of 1500 healthy and 1500 tumor-free brains, which it has previously trained on. The collection includes 3000 magnetic resonance pictures in total which was increased by data augmentation. When used to predict the presence of a tumor, the model produced great results, reaching 95.67 percent in validation data and up to 99.17 percent in test data.

Keywords - deep learning, convolutional neural networks, image classification, brain tumor detection.

ACKNOWLEDGEMENT

The successful completion of any task is incomplete and meaningless without giving any due credit to the people who made it possible without which the project would not have been successful and would have existed in theory.

First and foremost, we are grateful to **Dr. Dr. Md. Kamal Hosain**, HOD, Department of Electronics and Telecommunication Engineering, RUET, and all other faculty members of our department for their constant guidance and support, constant motivation and sincere support and gratitude for this project work. We owe a lot of thanks to our supervisor, **Md. Rakib Hossain**, Assistant Professor, Department of Electronics and Telecommunication Engineering, RUET for igniting and constantly motivating us and guiding us in the idea of a creatively and amazingly performed Major Project in undertaking this endeavor and challenge and also for being there whenever we needed his guidance or assistance.

We would also like to take this moment to show our thanks and gratitude to one and all, who indirectly or directly have given us their hand in this challenging task. We feel happy and joyful and content in expressing our vote of thanks to all those who have helped us and guided us in presenting this project work for our Major project. Last, but never least, we thank our well-wishers and parents for always being with us, in every sense and constantly supporting us in every possible sense whenever possible.

Contents

List of Figures

Chapter 1

INTRODUCTION

1.1 Overview

The brain can be affected by tumors, and they may target a specific region of the brain. This simultaneously and flawlessly kills those cells, damaging the brain. The damaged area of the brain and the size of the tumor mass determine the extent of the damage and how dangerous it is. Generally speaking, the brain communicates with the body's organs, blood, and nerve cells. Any brain impairment will have an impact on how the body functions normally and typically, causing the body to become unstable. When we talk about instability, we're talking about a person's consciousness, body balance, organ function, and mentality, which can occasionally have an impact on their health and physical state. Noting that some brain cell types do not undergo regeneration and that some cells cease to divide and undergo regeneration at a particular age in a person's life. One of the five senses can sometimes be lost when a tumor develops in a sensitive section of the brain, such as the region that regulates taste, touch, smell, and hearing. If the patient is fortunate enough to find the tumor at an early stage, medical intervention may involve surgery or radiotherapy. The possibility of a full recovery from the tumor following these operations cannot be ruled out with confidence. There are instances where one of the five senses experiences loss of feeling, memory loss, paralysis of one of the limbs, or all three. In addition, a portion of the brain that the cancer cells had nourished had been surgically or radiologically removed. It is necessary to highlight that brain tumors might be lethal in some circumstances since we discussed the conditions brought on by the tumor. One of these situations is the delay in tumor detection. As a result, the tumor has grown larger

and is now consuming more brain tissue, which is devastating for the patient's health and quality of life and may even be fatal.

An early diagnosis of the tumor is crucial. Known medical techniques, such as magnetic resonance imaging, are used to find tumors, and biopsies are also collected. We are interested in the MRI images since we will be feeding them to our model. The first diagnostic procedure for finding tumors is an MRI. The brain is cross-sectionally imaged using MRI, and the location and mass of the tumor are identified using individual image analysis. The difference in the degree of black-and-white contrast in the image allows one to identify the tumor from the photographs. The tumor appears as a brightly colored white block or pattern. Yet, certain regions of the brain exhibit the same traits as cancerous cells. Here, misdiagnosis may happen and play a part in medical procedures that rely on surgical techniques and biopsies for their operations. Thus, the patient must be aware of the tumor and be able to distinguish it from other brain regions.

Figure 1 depicts two MRI images for two brains, the one on the left is healthy and the one on the right is with tumor:

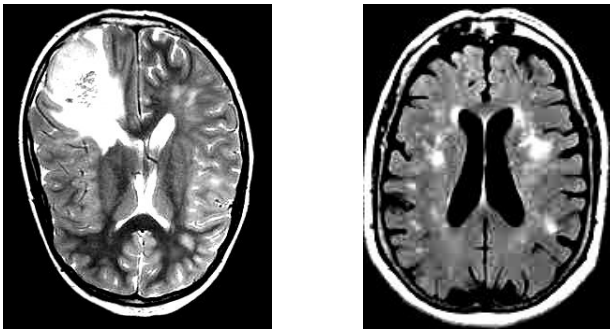

Figure 1.1: Imaging of two distinct brains using MRI. Tumor on the left, healthy on the right. [1].

Imaging techniques like ultrasound, tomography, magnetic resonance imaging and X-ray imaging are typically used to find illnesses and malignancies. The initial step in diagnosing illnesses and other ailments of the internal organs is imaging. Imaging provides a window into the body for medical professionals to discover and diagnose disease and injury. It necessitates that the person in charge of making a diagnosis based on images comprehends both normal and abnormal organ function conditions as well as the severity of any disease or damage.

Software and artificial models are being prepared to analyze and examine the MRI images

taken by the patient. The model is created and programmed to carry out image analysis by identifying and extracting the properties of an existing image. Since images are made up of an array of pixels with width and length, programs analyze them by using filters and mathematical techniques. Depending on the color system used (RGB, CMYK, Grayscale, etc.), each pixel represents a certain color and has a value for the color it represents. Each pixel has some sort of connection to its immediate neighbors. The pixels represent either a portion of an eye, an edge, or a box of some color. As you look at an image, it determines the objects in the image, just like the human eye does. To determine the pattern for these pixels, the model employs mathematical formulas and filters. After that, it extracts the image's attributes and gives them weights.

Figure 2 displays an original brain MRI picture on the left, a binary threshold filter applied in the middle, and a high pass filter applied to the right to identify edges:

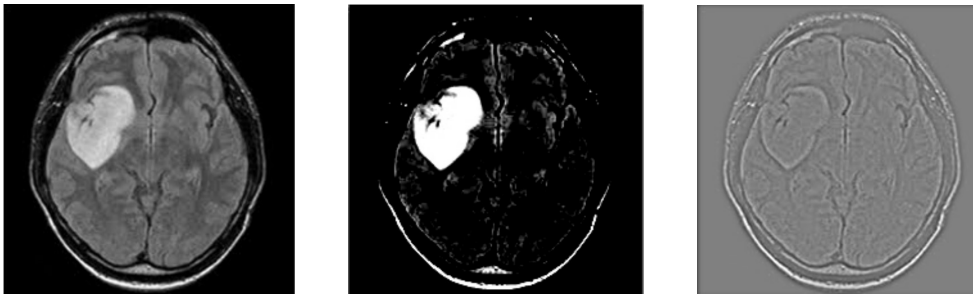

Figure 1.2: In order to extract properties, filters are applied to the MRI images [2].

Our objective is to develop a software program that utilizes an artificial convolutional neural network to analyze magnetic resonance imaging (MRI) images of the brain. This network will extract properties from the images and determine if a tumor is present. To accomplish this, we will input the images as a matrix of pixels arranged in rows and columns. Each cell in the matrix represents a pixel and stores a color value based on the color system in use. The neural network then processes the matrix in a layer by applying mathematical operations to assign meaning to the image. Next, the image is passed through a filter that removes irrelevant data and reduces the array's size. This process is repeated for each layer, based on its function and the number of layers involved. Finally, the neural network will assign labels to the properties and calculate the percentage of accuracy in identifying the correct labels.

For our model, we constructed a neural network using five Convolutional blocks, each with a different number of filters, to vary the scale between layers. Our model's output is

a percentage indicating the presence of a tumor. We used a dataset of MRI brain images that includes pictures of healthy brains and various types of tumors. The dataset contains 3000 images, of which 1500 depict a healthy brain and 1500 show a tumor. All images in the dataset are in grayscale. This dataset was obtained from the Kaggle website.

After programming and training the neural network on the dataset, we initially obtained near-satisfactory results in terms of tumor discovery in the images. This is due to what was mentioned earlier, that is due to the presence of properties of parts of the brain that have the same behavior of tumor cells during imaging so that they have the same appearance in the images. It was necessary for us to make certain adjustments to the neural network in order to overcome this obstacle. Indeed, we adjusted and prolonged the training period of the neural network, which would have yielded quite satisfactory results. Eventually, the neural network differentiated between tumors and parts of the brain with the same behavior. The neural network also identified tumors with an accuracy of 99 % in some images. The ratios were limited to images that were tested on the neural network between (70 % - 99 %) for images that contain tumors, and between (0 % - 25 %) for healthy pictures. The neural network was not that ideal either, as strange results appeared for images that were limited to between (40 % - 65 %). In the end, however, the neural network performed the required function with test accuracy up to 99.17 %.

After the model has been created, the whole model is wrapped up in a section and presented through a website. The website was created using simple tools such as HTML, CSS, bootstrap, javascript, python flask, etc, and is accessible to everyone worldwide.

1.2 Objectives

The sole purpose of the model, represented and accessed by a website is:

(1) to go through the MRI images thoroughly and extract the information through inspecting:

MRI images taken from patients are analyzed and examined through software and artificial models. These models are designed and programmed to identify and extract properties of images by analyzing an array of pixels with width and length, using filters and mathematical techniques. Depending on the color system used, each pixel represents a specific color and has a value associated with it. The model determines objects within the

image by analyzing the pattern of pixels and assigns weights to the image's attributes.

(2) to make a decision whether the images of the MRI scan of the brain has tumor cell or not with a specific amount of accuracy:
Our goal is to create a technique that uses an artificial convolutional neural network to examine MRI brain images and detect tumors. To achieve this, we will input the images as a matrix of pixels that hold color values according to the color system used. The neural network will then process the matrix layer by layer, using mathematical operations to interpret the image. A filter will be applied to eliminate irrelevant data and reduce the array's size. The process is repeated for each layer based on its function and the number of layers involved. Finally, the neural network will label the properties and calculate the accuracy percentage for identifying the correct labels.

1.3 Applications

A brain tumor detection website can have a variety of applications, including:

- **Early Detection:** A brain tumor detection website can be used to detect brain tumors in their early stages. This can help in starting the treatment process early, leading to better patient outcomes.
- **Efficient Diagnosis** The website can aid doctors in quickly diagnosing and treating the brain tumor. Early detection and efficient diagnosis can help reduce healthcare costs and improve patient outcomes.
- **Screening:** Brain tumor detection websites can also be used for screening purposes, where individuals can undergo routine scans to detect any abnormalities in their brain.
- **Patient Education:** The website can also educate patients and their families about brain tumors, their symptoms, and the treatment options available to them.
- **Remote Consultations:** A brain tumor detection website can enable remote consultations between doctors and patients, allowing patients to receive medical advice from the comfort of their homes.

- **Research:** The website can be used to collect data on brain tumors, which can be used for research purposes. This can help in developing new treatment methods and improving patient outcomes in the future.
- **Public Awareness:** The website can also be used to raise public awareness about brain tumors and the importance of early detection and treatment.

Chapter 2

Main Framework and Operations

2.1 Solution Framework:

To begin with, we will discuss the algorithm and mechanism of our neural network, which was designed with a focus on performance to create a lightweight network that could function optimally. Our design is based on the Artificial Convolutional Neural Network (ACNN), which includes an input layer for receiving tensors, five convolutional blocks, and a fully connected layer for linking the neurons together. Additionally, a dropout layer was added to reduce the load on the output layer, which contains two neurons with values of (0,1). To maintain its functionality, a different activation function was used in the last layer.

Before delving further into the details of the neural network, let's examine the flow chart of the entire process, as shown in Figure 2.1. The process begins with opening the images in grayscale mode to reduce complexity, followed by data augmentation to generate new images from existing ones. The images are then resized to a unified shape for input into the ACNN. Lastly, the training process is performed. In working with images, data cleaning may not be necessary, as we can replace it with data augmentation, which will be discussed later.

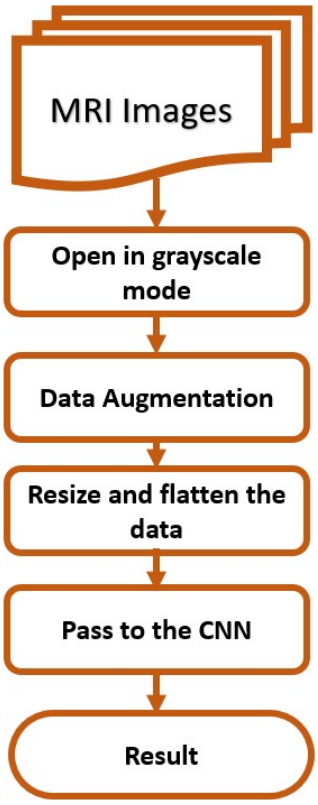

Figure 2.1: Flowchart of the procedure

2.2 Topology of CNN:

Many people focus on Convolutional Neural Networks (CNN) and regard them as excellent for image classification. A CNN is made up of nodes that perform mathematical operations such as multiplication, addition, entropy, distance, and more. The nodes are connected by data flow paths, and data in the form of tensors or matrices move between nodes and change shape based on each node's operation. But why are CNNs so effective? Computers don't see images the way we do. Instead, they view them as matrices of numbers, where each matrix has a width and height in terms of rows and columns. Each cell in the matrix represents a pixel, which is a unit of measurement for the resolution of the image based on the presence of color points in that area. CNNs handle images as matrices, but they focus on specific areas at a time rather than the entire image, allowing them to extract features from the image. Below is a brief demonstration of how CNNs handle images.

The figure below illustrates the basic structure of a convolutional neural network, demon-

strating how it observes an image and identifying the responsible components for this process:

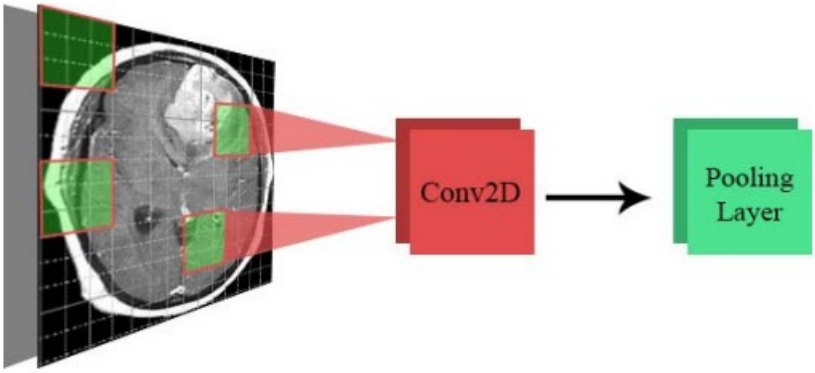

Figure 2.2: An overview of how convolutional layers function.

A two-dimensional convolutional layer employs a filter with a size of (n x n) to iterate over an image matrix. The filter performs mathematical operations between sectors and their values, generating a new matrix. After the convolutional layer processes the image matrix, a new matrix is produced and passed on to the pooling layer. The pooling layer segments the matrix and selects a specific value from each sector, depending on the type of pooling used, to reduce the values. The next figure illustrates the mechanism of the pooling layer using a 2x2 filter layer:

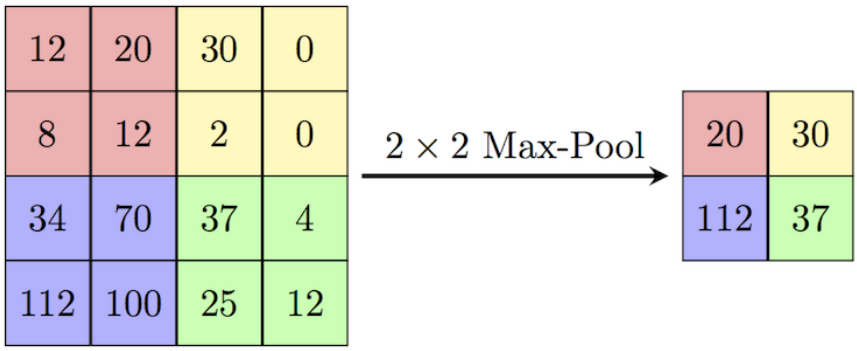

Figure 2.3: Working of max pooling.

The architecture of our ACNN is depicted in Figure 8. The types of layers that we have employed are described in the following ways:

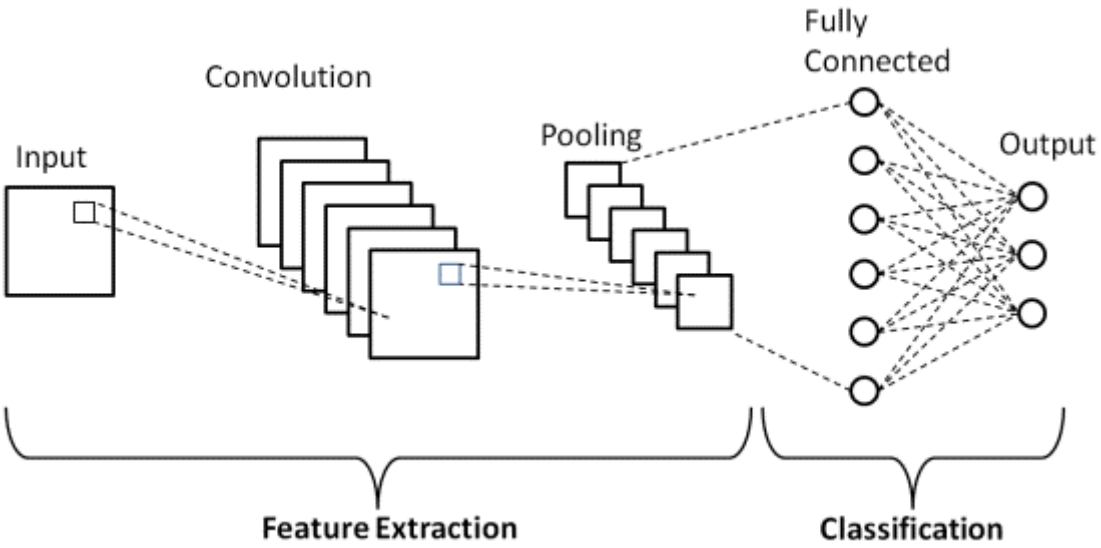

Figure 2.4: The design of our artificial convolutional neural network.

- **Input layer:** Its dimensions are (256, 256), and it will use photos as input for the ACNN.
- **Convolutional block:**We refer to each convolutional layer followed by a pooling layer as a convolutional block, however this is not a common term.
- **Convolutional layer (2D):**In order to create a pattern that the neural network will weight and learn, this layer will perform mathematical operations and filters to images while it is running. There is a filter and a filter size for each layer. The idea behind a filter is that it consists of a matrix with a set of values that will either be multiplied or used in other ways to manipulate the original image matrix.
- **Maxpool layer:**In general, layer pooling reduces the amount of values and dimensions across layers, which results in smaller tensors than before. There are other pooling techniques—we utilized MAX pooling—that will pull the greatest possible number from a given dimension.
- **Fully connected:**The amount of values and dimensions between layers is typically decreased by layer pooling, leading to smaller tensors than before. Some pooling methods—we used MAX pooling—will extract the largest number from a certain dimension.

- **Activation function:**is the switch that activates the layer in accordance with the incoming tensor from the preceding layer.

Using the same layer many times with the same settings is absurd! Due to the fact that we used five convolutional blocks, we tweaked each one to perform a unique function in comparison to the others, resulting in a strong, consistent, and coherent ACNN. Each 2D convolutional layer's filter count was adjusted. As a result, we used filters that were the right size, and we also constructed pooling layers that were the right size. An ACNN needs parameters to enhance execution and performance, lower the risk of losing control over the model, and keep the ACNN consistent and controllable. Two important factors are:

- **Loss function:**is a mathematical optimization function that converts a real number that reflects the "cost" of the event into the representation of an event. By shortening the distance between the points, a loss function aids in optimizing the ACNN's regression.

- **Optimizer function:**is the process of choosing the best element out of all the available options. In addition to the loss function, the optimizer function aids in regression and boosts the effectiveness of the ACNN.

As a loss function, we employed the "Categorical cross-entropy" function and the "Adam" optimizer function.

2.3 Coding Operations and Explanations

2.3.1 Importing libraries

```
import tensorflow as tf
from keras.preprocessing.image import ImageDataGenerator
```

Figure 2.5: Importing the libraries and modules

For the starting part, where we explored the dataset first and preprocessed it using the Image data generator, imported Tensorflow. In TensorFlow, we used the Keras library.

2.3.2 Data Preprocessing

```
datagen = ImageDataGenerator(
    rescale=1/255,
    validation_split=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True)
training_set = datagen.flow_from_directory(
    directory='dataset',
    subset='training',
    target_size=(64, 64),
    batch_size=32,
    class_mode='binary')
test_set = datagen.flow_from_directory(
    directory='dataset',
    subset='validation',
    target_size=(64, 64),
    batch_size=32,
    class_mode='binary')
```

Figure 2.6: Data preprocessing section

When we use any image dataset for training our model, sometimes it may not give us accurate results as the images in it may require some preprocessing like zooming, increasing brightness, changing the grayscale value, etc. Just like the binary data requires some data cleaning and preprocessing, the image dataset to needs it. For this, the Image data generator from the Keras library is used.

It generates batches of tensor image data with real-time data augmentation, e.g., resizing all the images, and adjusting their height and width, so that the input image data is uniform.

1. Args rescale: (rescaling factor). Defaults to None. Otherwise, we can multiply the data by the value provided.

2. Shear_range: Shear Intensity (Shear angle in a counter-clockwise direction in degrees)

3. Zoom_range: Range for random zoom

4. Height_shift_range: fraction of total height, if less than 1

5. Width_shift_range: fraction of total width, if less than 1

6. Fill_mode: Default is 'nearest'. Points outside the boundaries of the input are filled according to the given mode.

7. Validation_split: Fraction of images reserved for validation (strictly between 0 and 1).

Some arguments used here are,

- Target_size: Tuple of integers (height, width), defaults to (512, 512). The dimensions to which all images are found will be resized.
- Seed: Optional random seed for shuffling and transformations.
- Batch_size: Size of the batches of data (default: 32).
- Subset: Subset of data ("training" or "validation").
- Shuffle: Whether to shuffle the data (default: True) If set to False, sorts the data in alphanumeric order.

2.3.3 CNN Layers

Convolution

```
cnn = tf.keras.models.Sequential()
cnn.add(tf.keras.layers.Conv2D(filters=32, kernel_size=3, activation='relu',input_shape=[64,64,3]))
cnn.add(tf.keras.layers.MaxPool2D(pool_size=2, strides=2))
cnn.add(tf.keras.layers.Conv2D(filters=32, kernel_size=3, activation='relu'))
cnn.add(tf.keras.layers.MaxPool2D(pool_size=2, strides=2))
cnn.add(tf.keras.layers.Flatten())
cnn.add(tf.keras.layers.Dense(units=128, activation='relu'))
cnn.add(tf.keras.layers.Dense(units=1, activation='sigmoid'))
cnn.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
history = cnn.fit(x = training_set, validation_data= test_set, epochs=25)
✓ 0.2s
```

Figure 2.7: Implimenting the CNN layers

Neural networks are widely used for almost every deep learning project because of their accuracy and ability to detect data without being explicitly programmed. Different kinds of neural networks are used according to the project's needs; for example, we will use Artificial Neural Networks (ANN) for integer data. CNN is widely used for classifying

image data. The main advantage of CNN is that it automatically detects important features in any image without human supervision. This may be why CNN would be a perfect solution to computer vision and image classification problems. Feature extraction is therefore vital for CNNs.

The performed feature extraction consists of three basic operations:

1. Filter an image for a particular feature (convolution)
2. Detect that feature within the filtered image (ReLU)
3. Condense the image to enhance the features (maximum pooling)

Convolution

The convolutional layer performs the filtering step. The weights that ConvNet learns during training are primarily contained in its convolutional layers. These scales are called cores. The core works by scanning the image and producing a weighted pixel sum.

Activation Function

This is the most important part of a neural network. The activation function decides whether a particular neuron will fire or not based on the input it receives and passes it on to the next layer. The Rectified Linear Unit or ReLU is the most common activation function used due to its simple implementation and overcoming many other obstacles caused by other activation functions such as the Sigmoid.

We also used the Sigmoid activation function in the model because it is used to classify multi-class datasets.

Maxpooling

Max Pooling is a convolutional process where the kernel extracts the maximum value of the area it covers. Like max pooling, we can use average pooling. The ReLU (Detect) function in the function map ends up with a lot of "dead space", we would like to condense the function map to keep only the most useful part of the function itself.

2.3.4 Visualization and Model Creation

```
cnn.save('BrainTumor25Epochs.h5')
✓ 17.8s

import matplotlib.pyplot as plt
epochs = range(len(history.history['accuracy']))
plt.plot(epochs, history.history['accuracy'], 'green', label='Accuracy of Training Data')
plt.plot(epochs, history.history['val_accuracy'], 'red', label='Accuracy of Validation Data')
plt.xlabel('Total Epochs')
plt.ylabel('Accuracy achieved')
plt.title('Training and Validation Accuracy')
plt.legend(loc=0)
plt.figure()
✓ 1.9s
```

Figure 2.8: Model creation and visualization section

the first section is used to create a model of the code and used it in further applications i.e., in the website.

We will plot a graph between our accuracy achieved and the number of epochs to see how our training and validation accuracy increases. Result of the last two epochs are given bellow:

```
Epoch 24/25
75/75 [==============================] - 17s 225ms/step - loss: 0.0191 - accuracy: 0.9942 - val_loss: 0.0496 - val_accuracy: 0.9817
Epoch 25/25
75/75 [==============================] - 16s 208ms/step - loss: 0.0300 - accuracy: 0.9917 - val_loss: 0.1396 - val_accuracy: 0.9567
```

Figure 2.9: Ultimate result of the last two epochs

Chapter 3

Design & Implementation of the Project

3.1 Website's Visual Representation Portion:

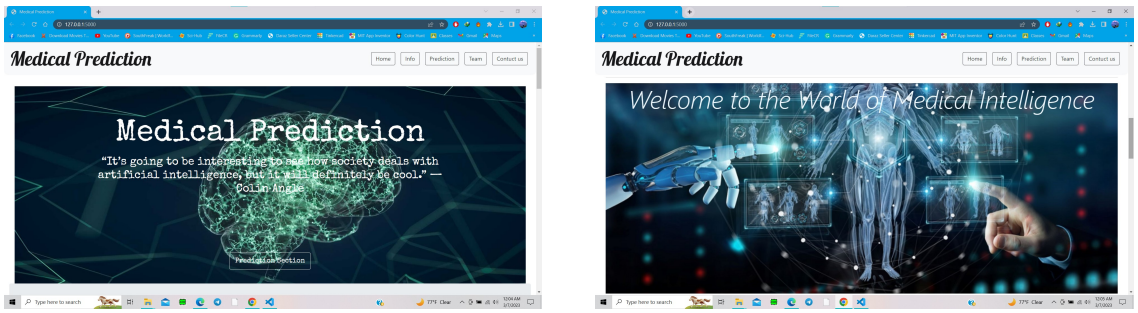

Figure 3.1: Visual Representation Portion of the Website.

A website's visual representation generally includes a combination of text, images, videos, and other multimedia elements arranged in a visually appealing and organized manner. The website's layout and design are usually based on a specific theme or brand identity that reflects the website's purpose and target audience.

The website's header usually contains a logo or title, a navigation menu that allows users to browse different pages, and sometimes a search bar to find specific information. The body of the website is where the main content is located, and it may include sections such as an introduction, features, pricing, testimonials, and a call-to-action.

3.2 Information Section:

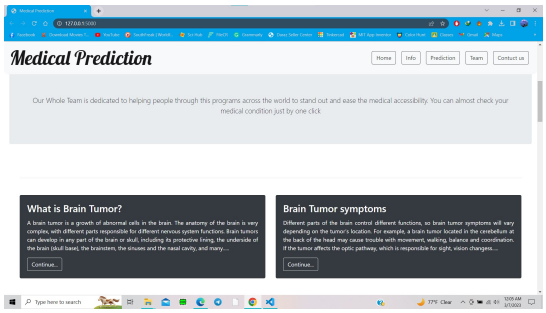

Figure 3.2: Information Section of the Website.

The information section of a website typically contains detailed information about the website's purpose, products, services, or other relevant topics. This information section includes an introduction to the website, an overview of the company or organization behind the website, and a mission statement or values statement that explains what the website is all about. It also includes detailed information about the brain tumor cell, symptoms of the existence of brain tumor cells, and their precautions This section also include a blog or news section, which provides users with the latest updates and insights about brain tumor cell treatments.

3.3 Prediction Section

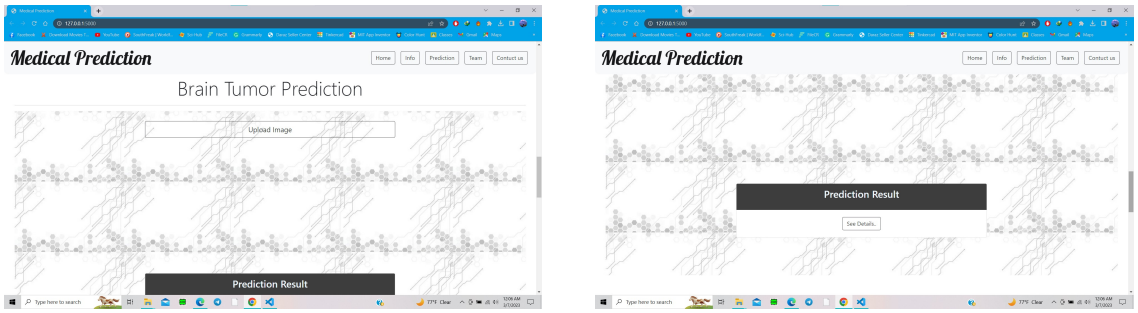

Figure 3.3: Prediction Section of the Website.

The prediction section typically uses advanced algorithms and machine learning techniques to predict the likelihood of a brain tumor based on medical images or other diagnostic data. Users typically upload or provide access to their medical images or diagnostic

data, which are then analyzed by the website's prediction algorithm. The prediction section may ask users to specify certain parameters or characteristics of the images, such as the size or location of the tumor, to help refine the prediction. in this case, it requires MRI images to do the required operation.

Once the analysis is complete, the prediction section will provide users with a prediction of the likelihood of a brain tumor. The prediction also includes a confidence level or a probability score that reflects the algorithm's level of certainty about the prediction.

The prediction section is designed to be user-friendly and easy to understand, with clear explanations of the prediction and any relevant medical terminology or concepts. at last the user's privacy is well preserved.

3.4 Team Description and Communication Information Section:

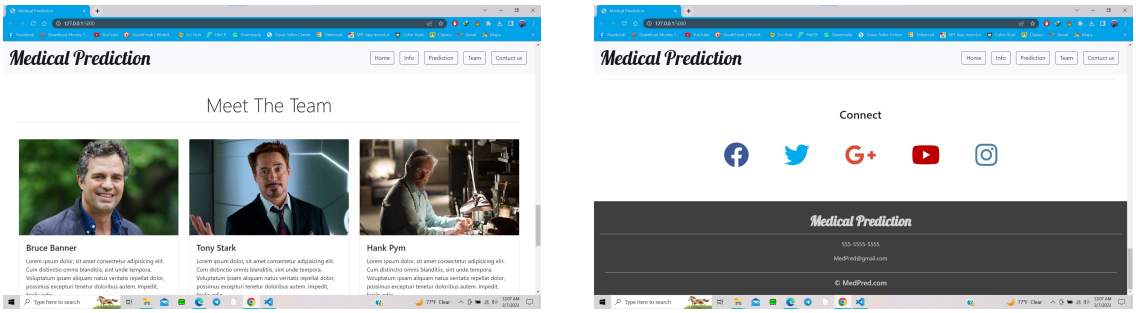

Figure 3.4: Team description and communication information section of the Website.

The team and communication section of a website typically provides information about the people behind the website or organization, as well as how users can communicate with them.

The team section usually includes information about the website's founders, executives, or key personnel, along with their biographies, photos, and areas of expertise. This section may also include testimonials from satisfied customers or partners, which can help build trust and credibility with potential users.

The communication section provides users with various ways to contact the website's team, such as email, phone, or live chat. It may also include a contact form that users can fill out to submit questions or inquiries. This section may also provide information about

the website's customer support policies, such as hours of operation, response times, or frequently asked questions.

3.5 Overall Structure of the Website

Overall the modern, responsive, and dynamic website that provides a seamless user experience is created using these programming languages and the help of their modules and libraries:

- **HTML:** HTML is used to structure the content of the website, including text, images, and other multimedia elements. The HTML code defines the basic layout of the website and provides a structure for the content to be displayed.
- **CSS:** CSS is used to style the website's appearance, including colors, fonts, and layout. The CSS code can be used to create a visually appealing and cohesive design that reflects the website's brand identity and purpose.
- **JavaScript:** JavaScript is used to add interactivity and functionality to the website, such as animations, user input validation, and dynamic content updates. JavaScript can be used to make the website more user-friendly and engaging, enhancing the user experience.
- **Flask:** Flask is a Python web framework that is used to create dynamic web applications. It provides tools and libraries for building web applications, handling user requests, and serving dynamic content. Flask can be used to create back-end functionality that interacts with the website's front-end components, allowing for seamless communication between the two.

Chapter 4

Results Analysis & Discussion

Testing is a crucial aspect that determines the success or failure of your work, and this is especially true for the creation of ACNNs. The testing process evaluates the ACNN and assesses the accuracy of its predictions as well as its ability to predict the correct label. Testing also plays a key role in detecting problems within the ACNN, such as overfitting. To carry out testing, we used 80% of the data for training and reserved 20% for testing. Over a period of two weeks, we conducted extensive testing on our ACNN, experimenting with different tuning methods to identify the most suitable structure and level of accuracy. This involved testing the ACNN with various optimizer and loss functions, as well as adjusting layer parameters such as filter numbers and sizes.

Table 4.1 displays the highest accuracy rates and loss values achieved by our ACNN model after being trained for 25 epochs. Additionally, some of the test cases yielded an accuracy score of 99.99% in detecting cancer, and we categorized the model outputs into three main groups: no tumor range, outliers range, and tumor range, with percentages ranging from 0% to 25%, 40% to 65%, and 70% to 99%, respectively. It is important to note that misclassification cases may occur in any model, and our model was no exception. We observed instances of misclassification, which were attributed to certain brain components in magnetic resonance exhibiting similar characteristics to tumor masses and

Table 4.1: Testing Data

Formation	***Accuracy***	***Loss***
Testing	99.17%	3%
Validation	95.67%	13.96%

cells. Graphical representation of the test result is given bellow:

Figure 4.1: Accuracy representation of the model.

On the website a user can see the results as follows:

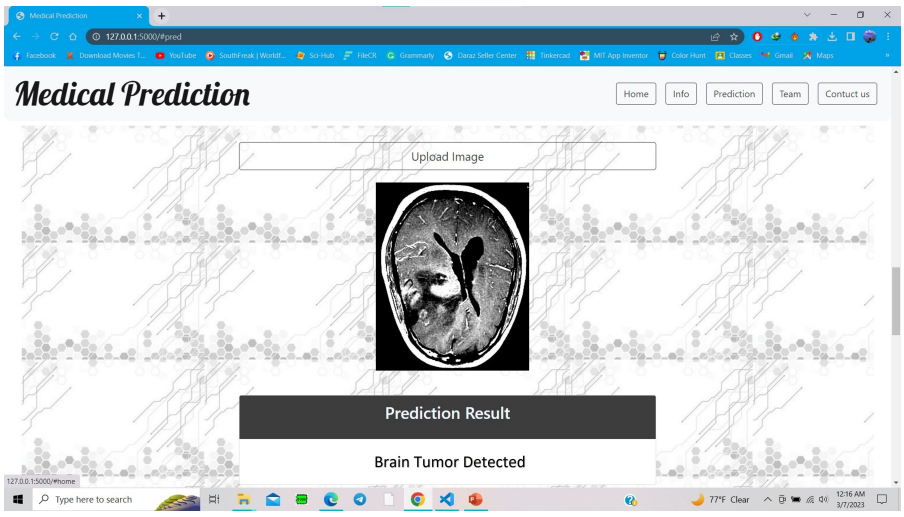

Figure 4.2: Demo representation at the website.

Chapter 5

CONCLUSION

Tumors pose a fatal threat to humanity, causing numerous mortalities and disabilities globally. One of the most deadly tumor types is a brain tumor, which impacts the nervous system, blood system, and organ function. Various methods are available to detect tumors, with one utilizing MRIs to produce cross-sector images of the brain, allowing for better detection of affected areas. In our research paper, we utilized a dataset provided by Kaggle consisting of 3000 MRI images for brains, with 1500 for brains with tumors and 1500 for healthy brains. We increased the dataset size using augmentation and designed an ACCN to classify MRI images as either positive (having a tumor) or negative (not having a tumor). By experimenting with our ACNN on a test set, we achieved a 95.2507% accuracy rate in identifying tumors. Our Artificial Convolutional Neural Network accurately predicts brain tumors from MRI images of the brain, with a test accuracy rate of 99.17% and an evaluation accuracy rate of 96.67%. We plan to further refine our ACNN by tuning its settings and training it on more data from other sources or through additional augmentation of available images.

.

Bibliography

[1] "Brain tumor code description," ://www.analyticsvidhya.com/blog/2022/07/building-a-brain-tumor-classifier-using-deep-learning/, 2022.

[2] https://en.wikipedia.org/wiki/Braintumor , November, 21 2019., 2022, Accessed on August.

Appendix

Brain Tumor Detection Code:

```
    import tensorflow as tf
from keras.preprocessing.image import ImageDataGenerator

## data preprocessing
datagen = ImageDataGenerator(
    rescale=1/255,
    validation_split=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True)
training_set = datagen.flow_from_directory(
    directory='dataset',
    subset='training',
    target_size=(64, 64),
    batch_size=32,
    class_mode='binary')
test_set = datagen.flow_from_directory(
    directory='dataset',
    subset='validation',
    target_size=(64, 64),
    batch_size=32,
    class_mode='binary')

# convolution layers
cnn = tf.keras.models.Sequential()
```

```
cnn.add(tf.keras.layers.Conv2D(filters=32, kernel_size=3, activation='relu',input
cnn.add(tf.keras.layers.MaxPool2D(pool_size=2, strides=2))
cnn.add(tf.keras.layers.Conv2D(filters=32, kernel_size=3, activation='relu'))
cnn.add(tf.keras.layers.MaxPool2D(pool_size=2, strides=2))
cnn.add(tf.keras.layers.Flatten())
cnn.add(tf.keras.layers.Dense(units=128, activation='relu'))
cnn.add(tf.keras.layers.Dense(units=1, activation='sigmoid'))
cnn.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
history = cnn.fit(x = training_set, validation_data= test_set, epochs=25)

#save the model
cnn.save('BrainTumor25Epochs.h5')

#visual representation of the result
import matplotlib.pyplot as plt
epochs = range(len(history.history['accuracy']))
plt.plot(epochs, history.history['accuracy'], 'green', label='Accuracy of Trainin
plt.plot(epochs, history.history['val_accuracy'], 'red', label='Accuracy of Valid
plt.xlabel('Total Epochs')
plt.ylabel('Accuracy achieved')
plt.title('Training and Validation Accuracy')
plt.legend(loc=0)
plt.figure()
```

Code of Applying the Module to the Website(Back-end):

```
from flask import Flask, render_template, request
from werkzeug.utils import secure_filename
import os

app = Flask(__name__)


@app.route('/', methods = ['GET'])
def index():
   return render_template('index.html')

def model_predict(img_path):
   from keras.models import load_model
   from keras.preprocessing import image
   from PIL import Image
   import numpy as np
   import tensorflow as tf
   import cv2

   model = load_model('BrainTumor25Epochs.h5')
   image = cv2.imread(img_path)
   img = Image.fromarray(image)
   img = img.resize((64,64))
   img = np.array(img)
   input_img = np.expand_dims(img, axis=0)
   pred = model.predict(input_img)
   pred = int(model.predict(input_img) )

   return pred

@app.route('/predict', methods = ['GET', 'POST'])
```

```
def upload():
    if request.method == 'POST':
        f = request.files['file']
        #saving image to directory
        basepath = os.path.dirname(__file__)
        file_path = os.path.join(
            basepath, 'uploads', secure_filename(f.filename))
        f.save(file_path)
        result = " "
        # Make prediction
        r = model_predict(file_path)
        if r == 1:
            result = "Tumor Cell Detected"
        if r == 0:
            result = "No Tumor Cell Detected"
        return result
    return None


if __name__ == '__main__':
    app.run(debug = True)
```