Automating Brain Tumor Classification with the Use of Convolutional Neural Networks

Lagnajeet Panigrahi

Shrewsbury High School

Abstract

Every second could mean the difference between the life and death of a patient with a brain tumor. Delays and misinterpretations from radiologists will affect the administration of treatment, which could have consequences for patients. It is quintessential to consider new methods of diagnosis that will minimize fallacies present in diagnosis and will maximize speed. This project attempted to solve this problem by using artificial intelligence, in particular, convolutional neural networks (CNNs). The goal was to create a CNN that is capable of distinguishing—using MRI scans— glioma, meningioma, pituitary tumors, and brains that do not have any tumors. Upon completion of the CNN and the analysis of the data, it was deduced that this program's accuracy was between 87.5% to 91.67%. This project indicates that, after more refinement, CNNs can be used in medical environments for diagnosis. At minimum, they can be used as an aid to radiologists.

Research Paper

Every second could mean the difference between life and death for a patient when it comes down to diagnosing and classifying brain tumors. Delays and misinterpretations of data will slow the speed of the administration of treatment, which could have disastrous consequences on a patient, making it all the more important to reach conclusions as fast and accurately as possible (Onder et al., 2021). Currently, brain tumors are being classified by radiologists by performing an MRI (Magnetic Resonance Imaging) scan and then manually analyzing the resulting image. Although this method is viable, it is susceptible to many errors. In fact, in a study, it was found that in the practice of radiology, errors are more common than they should be: it is estimated that the day-to-day rate of error is 3-5% of reported cases (Brady, 2017). Not only is the traditional method prone to mistakes, but it also creates delays in diagnosis due to the length of the process radiologists perform (National Cancer Institute [NCI] Staff, 2020). Consequently, these problems result in a system that fails to perform in a manner that is most beneficial to the lives of the patients. To overcome this, a new system is required: one that can diagnose life-threatening brain tumors earlier to ensure the timeline for treatment is not compromised whilst also maintaining accuracy. One proposed method of achieving this is through the utilization of artificial intelligence (AI). Studies show that certain models of AI can diagnose brain tumors in significantly less time than radiologists (NCI Staff, 2020). Ultimately, this project strives to use AI, in particular, convolutional neural networks created in Python, to accurately diagnose and categorize brain tumors in an effort to remove human error and shorten delays. The program's success will be determined by how accurately and consistently it can interpret the MRI scans. A project like this will greatly impact the scientific community as it will not only provide a method to automate brain tumor classification but, pave the way for the use of

AI for the diagnosis of other health-related problems. This project will build on previous projects that have used machine learning algorithms by instead using deep learning, allowing the program to be more accurate when it comes down to the classification and recognize images.

Currently, there is an immense problem surrounding the diagnosis of brain tumors: the speed and possible misreadings. Misinterpretations can be blamed for biases and interpretation discrepancies between radiologists (Onder et al., 2021). Brain tumors can be either fast-growing or slow-growing (Cleveland Clinic, 2022). Although slow-growing tumors are less severe, they can still cause a multitude of problems by growing large enough to press on surrounding nerves, blood vessels, and tissue (Cleveland Clinic, 2022). Thereby, it is imperative to have a method to remove these differences as they pose major clinical consequences for the patients (Ondor et al., 2021) Additionally, in a study, 4% of the sampled brain tumor cases could have been diagnosed at an earlier date but were overlooked by the radiologist (Stempniak, 2022). This proves that there is an existing problem regarding the diagnosis of brain tumors that needs resolving for the betterment of the patients. To mitigate these problems, this project will use AI as it can provide one consistent platform for diagnosis and provide a multitude of additional benefits such as faster diagnosis. In 2019, a project similar to this was created to identify pneumonia based on chest scans (Yadav, 2019). This program yielded a 96.6% accuracy rate for the classification of pneumonia (Yadav, 2019). This proves that AI can be a viable method of diagnosis.

In order to create a program that can correctly identify and understand brain MRI scans, it is essential to know what brain tumors are, how they occur, and the different types of brain tumors and their characteristics. Brain tumors are masses or the growth of abnormal cells inside an individual's brain (Mayo Clinic, 2021). These tumors can be life-threatening and cause many unwanted symptoms such as headaches, seizures, vision problems, mental changes, difficulty with walking and speaking, and many more (Mayo Clinic, 2021). Brain tumors begin with normal cell changes and mutations in a cell's DNA (Mayo Clinic, 2021). Mutations are not always detrimental to one's health, but alterations in the DNA of the cell have the probability of causing a brain tumor (Cafasso, 2021). Specifically, mutations in the proto-oncogene, the gene that controls cell growth and division, and tumor suppressor genes can cause them to become hyperactive and accelerate cell division within the brain (Cafasso, 2021). Point mutations, gene amplifications, and chromosomal translocation can cause changes in the proto-oncogene (Healthline, 2022) The irregular emergence of unnecessary cells in the brain that could have characteristics of cancer cells such as uncontrollable cell division and inability to undergo apoptosis is the culprit for brain tumors (Eldrige, 2022). Apoptosis is vital as it allows for the removal of unwanted cells that are malfunctioning and plays a major role in deterring cancer through programmed cell death (National Human Genome Research Institute, 2022). There are various different types of brain tumors that can occur. This project will focus on identifying and differentiating three different types of brain tumors: meningioma, glioma, and pituitary tumors. Meningioma tumors are the most common type of primary brain tumor, making up thirty percent of all brain tumor cases (John Hopkins Medicine, 2022). These brain tumors originate in the meninges; the three layers of membranes that protect the brain and spinal cord (John Hopkins Medicine, 2022). Eighty-five percent of meningioma tumors are slow-growing benign tumors (John Hopkins Medicine, 2022). Gliomas are another type of common brain tumor that originates in the brain and can sometimes be found in the spinal cord (John Hopkins Medicine, 2022). These brain tumors stem from glial cells that support and surround neurons (John Hopkins Medicine, 2022). Almost all glioma tumors are malignant, but there are exceptions where they can be benign (John Hopkins Medicine, 2022). Pituitary tumors originate from abnormal growth within the small pituitary gland in the brain (John Hopkins Medicine, 2022). Most pituitary tumors are benign but still can cause many problems throughout the body as the pituitary gland creates hormones that affect functions in the human body (John Hopkins Medicine, 2022). There are multiple subcategories of each type of tumor, but, this project will focus on diagnosing the type of tumor instead of also focusing on the nuances within the categories. The classification of all of the subcategories of brain tumors that exist can be explored in future iterations and implemented at a later date.

There are multiple ways brain tumors are being diagnosed. For this project, MRI scans will be utilized along with AI. Magnetic resonance imaging (MRI) uses a magnet, radio waves, and a computer to create multiple detailed pictures that can be used to see brain tumors (Dana-Farber Cancer Institute, 2022). MRI scans are one of the best ways to diagnose brain tumors as they output more detailed results than other mediums of diagnosis such as CT scans (Cancer.net, 2021). There are three different orientations of MRI brain scans: Axial, coronal, and sagittal scans (Sriramakrishnan, 2020). Axial images provide a top to down view, sagittal images show a side-to-side view, and coronal scans give a front-to-back view (Sriramakrishnan, 2020). Using these images, it is possible to identify brain tumors and their location in the brain and in the MRI scan. Based on the location of the brain tumor and how the tumor appears on the scan, it is possible to determine the type of brain tumor. Meningioma brain tumors originate from the meninges: the outer three layers of tissue between the skull and brain (John Hopkins Medicine, 2022). This means these tumors can be found in places such as the surface under the skull (convexity meningioma), in the sphenoid wing (sphenoid wing meningiomas), on the underside of the brain (petrous meningioma), and many more areas (John Hopkins Medicine, 2022). On MRI scans, these brain tumors appear as extra-axial masses with a broad dural base

(Fahrenhost-Jones, 2022). This means that they are located out of the brain parenchyma and are near structures that line or surround the brain (Fahrenhost-Jones, 2022). On MRI, these tumors also tend to appear to be homogenous and well-circumscribed with rarer cases varying (Fahrenhost-Jones, 2022). Gliomas—tumors originating within the brain rather than near the outermost layers of the brain— can often be found in the cerebrum and cerebellum of the brain (John Hopkins Medicine, 2022). The appearance of gliomas on MRI scans varies more than meningiomas. Gliomas are intra-axial tumors which means they are located within the brain parenchyma and arise from brain cells (John Hopkins Medicine, 2022). Depending on the type of glioma, on MRI, it can appear to be a dark area compared to the rest of the brain, a dark area circumscribed by a contrasting outside (black on the inside and white on the perimeter), or have a mostly white appearance with little to no darkness in the center (MRI Appearance of Primary Brain Tumors, n.d.). Pituitary tumors occur in the pituitary gland which is located behind the back of the nose (John Hopkins Medicine, 2022). Although an axial MRI scan can be used for the diagnosis of a pituitary tumor, sagittal and coronal scans better show the tumor and thereby are the preferred orientation to use (Khan et al., 2022). Pituitary tumors on MRI scans are closer to the eyes and are hyperintense compared to the rest of the brain (Khan et al., 2022).

This project aims to use AI to automate the diagnosis of brain tumors. Deep learning is part of machine learning, which is a subset of AI. Deep learning develops algorithms that analyze data using a logical structure that mirrors human decision-making (Wolfewicz, 2022). These algorithms have the capability of performing various tasks such as the classification of data, anomaly detection, speech recognition, audio generation, image processing, and numerous more (Wolfewicz, 2022). For the fast and accurate diagnosis of these tumors, this project will employ deep learning, specifically, convolutional neural networks (CNN's). Convolutional neural networks are developed to function with grid-structured inputs such as two-dimensional images (Aggarwal, 2018). As a result of this, CNN's can be utilized for the classification of brain tumors using their image recognition capabilities to interpret MRI scans. Similar to how a human analyzes an image, a convolutional neural network takes an image and divides it into smaller sub-images (Lang, 2021). Then each sub-image is examined one by one and assembled to process and interpret the image (Lang, 2021). A neural network is created by the construction of several interconnected nodes organized into the input layer, hidden layers, and output layer (Kadlaskar, 2021). During the input and hidden layers phase, the neural network will perform a series of convolutions and pooling operations during which features of the inputted image are detected (Cornelisse, 2018). The goal of the convolutional layer is to reduce the dimensions of the image by extracting features based on filters that have been created, making it easier to interpret (Lang, 2021). The pooling layer strives to reduce the dimensionality of the image further and take out the relevant sections of the picture (Lang, 2021). Additionally, this layer filters out background elements of the image that do not contribute to the classification (Lang, 2021). The output part of the CNN, using the fully connected layers, will serve as a classifier for the extracted and compiled features (Cornelisse, 2018). This will output the prediction of the algorithm of the given object in the inputted image (Cornelisse, 2018). The percent accuracy of this program will be calculated by dividing the number of correct predictions by the total number of MRI scans analyzed (Bressler, 2021).

This project is fully dependent on the usage of the programming language Python and data sets for the creation of CNN's. As a result, this project utilizes multiple applications required to code in Python. There are many integrated development environments that can be used for this project such as Visual Studio Code and Pycharm. Instead of using a conventional

(IDE) to code this project, Jupyter Notebooks will be utilized. Jupyter Notebook is an open-source web application used to share and create documents that can include live code, visualizations, equations, and text (Driscoll, n.d.). Advantages of Jupyter Notebook include being free to use to the public, user-friendly, and easier to convey the entire process of a project to the programmer and audience. Jupyter Notebook will be installed via Anaconda which is a single streamlined package that contains and gives access to the current versions of Python and various libraries such as Pycharm (Domino, n.d.). Next, this project also requires the importing of TensorFlow and Keras libraries. Keras operates on top of the TensorFlow platform. TensorFlow is an open-source library created by Google with the main objective of creating deep learning applications (Simplilearn, 2022). Numpy will also need to be installed for the creation of CNN's. Numpy is a library that adds support for arrays, matrices, and other mathematical functions. This will aid in the creation of all of the layers needed within the neural network (W3Schools.com, n.d.). Finally, OpenCV is required to be installed. OpenCV is a tool that supports programming languages including Python, Java, and C++, and is great for image processing and computer vision tasks (Gupta, 2022). This allows it to be used in the context of the project. Data sets are also a key part of creating any neural network. There must be a data set for training the program to identify brain tumors and a data set for testing the accuracy. Data sets of brain tumor MRIs that are used in this project will be from Github. They will contain brain MRI scans of a healthy brain, pituitary tumors, meningioma tumors, and glioma tumors. The scans will be divided into two folders: one for training and one for testing.

The main objective of this computer science project is to use CNN's to increase the speed and accuracy of diagnosis as well as remove the inconsistencies between radiologists to create a more favorable method of diagnosing brain tumors. The creation of these tools will undoubtedly lead to the saving of lives as it will allow for a faster and more accurate diagnosis, ensuring that appropriate treatment for the patient is delivered in a timely manner. This deep learning program will first identify and confirm that there is a brain tumor, then proceed to classify the brain tumor as meningioma, glioma, or pituitary using a convolutional neural network throughout the whole process. Then based on what tumor is identified, it will convey the probability of the tumor being malignant. This project will be tested for speed of analysis and accuracy of diagnosis. As a whole, this project will bring automation to the health industry in a manner that is both effective and efficient.

Experimental Design

Location

This experiment will be executed indoors at home or at school with access to the internet. It will take place on a 15.6-inch HP Omen laptop. This laptop is equipped with an i7-9750H processor, an NVIDIA GTX 1660ti, 16 gigabytes of ram, 256 gigabytes of SSD storage, and 1 terabyte of hard drive storage.

Constraints

Since this experiment is solely based on programming, there are no independent variables, dependent variables, or controls to consider. Instead, this project will be tested on the accuracy of the created program. However, one major constraint to take into account is time. This is due to the fact that it is time-consuming to both develop the program itself and then test it to improve the program's accuracy based on results. Another limitation that needs to be addressed is the amount of data that can be used to create this program. Many programs related to artificial intelligence—such as the neural networks in this project— require a training dataset to allow the program to learn how to identify what is being shown. A testing set is also required to calculate the accuracy of this program and to ensure that the program is functioning correctly. This creates a great demand for datasets with various types of brain tumors. Since datasets for brain tumors are not widely available, the program will not be able to use the most optimal amount of data, thereby unable to reach its full potential. Overall, given more time and data, the program would be more accurate.

Safety

This entire project revolves around using a computer to program a convolutional neural network. The testing and training will also occur on a laptop. Therefore, there are no safety concerns to consider for the execution of this experiment.

Materials

1. In this experiment, a laptop equipped with an Intel i7-9750H processor, an NVIDIA GTX 1660ti, 16 gigabytes of RAM (random access memory), 256 gigabytes of SSD (solid state drive) storage, and 1 terabyte of hard drive storage will be utilized to perform the testing and creating of the program. The minimum recommended specifications for a computer that will be utilized for deep learning should have at minimum 8 or 16 gigabytes of random access memory, 256 or 512 gigabytes of SSD storage, and generally, a CPU (central processing unit) that is fairly new (i.e. 8th generation intel CPU and onwards).
2. There are many applications that need to be installed to begin programming a neural network. The following will be installed/imported (see procedure for instructions):
    a. Anaconda (An open-source distribution of Python that provides access to libraries and Jupyter Notebook)
    b. Jupyter Notebook (Jupyter Notebook is an open-source web application used to share and create documents that can include live code, visualizations, equations, and text. In this project it will be used to code the project)
    c. Tensorflow/Keras libraries (TensorFlow is an open-source library created by Google with the main objective of creating deep learning applications. Keras works on top of the Tensorflow platform)

  d. Numpy (a library that adds support for arrays, matrices, and other mathematical functions)
  e. OpenCV (a tool that supports programming languages including Python, Java, and C++, and is great for image processing and computer vision tasks)
  f. Matplotlib (this is a graphical plotting and data visualization library for python and Numpy)

3. The testing and training datasets will be obtained from GitHub via Kaggle (2020): https://github.com/sartajbhuvaji/brain-tumor-classification-dataset. These will contain brain MRI scans of a healthy brain, pituitary tumors, meningioma tumors, and glioma tumors. The scans will be divided into two folders: one for training and one for testing.

Procedure 1- Downloading and Importing necessary software

1. Installing Anaconda
   a. To begin, go to the Anaconda download page on the computer: https://www.anaconda.com/products/distribution.
   b. Select the operating system of the computer (windows, Linux, or macOS).
   c. Download the most recent python release. Remember to select either the 64-bit or 32-bit version depending on if a 64-bit or 32-bit version of Windows is being used.
   d. Once the installer has completed downloading, open and run the Anaconda installer.
   e. On the first screen that appears after running the installer, click “Next” to confirm the installation.
   f. Agree to the license agreement

g. On the next page, it is suggested that you click “Just Me”, which will install Anaconda on the current user account on the laptop. Select “All Users” only if it needs to be installed for all the users on the laptop (this will require Windows Administrator privileges). Click “Next” after the installation has been chosen.

h. Select a destination folder to install Anaconda and click “Next”

i. Select whether to add Anaconda to the laptop’s PATH environment variable or register Anaconda as your default Python. It is not recommended to add Anaconda to a PATH environment variable, since this can interfere with other software.

j. Click “Install”

k. After the installation click “Next”

l. The final screen will show a success message: “Thanks for installing Anaconda”. This means Anaconda has successfully been installed

2. Creating a new environment for the installation of libraries

a. Open the previously installed Anaconda Navigator

b. On the left-hand side click the tab that says “environments”

c. On the bottom of the environment page, click “create”

d. Name the environment tensorflow

e. Choose python 3.10.6

f. Click “create” to successfully create a new environment

g. To access Jupyter notebooks on this new environment, return to the home page and select “all applications” as well as the newly created environment. Next, click the install button on Jupyter notebook to gain access to the notebooks for coding purposes

3. Importing TensorFlow & Keras
    a. First, open the Anaconda navigator that was previously installed
    b. Go to the environments tab on the left-hand side
    c. Click on the newly created environment from step 2
    d. On the right-hand side, a search bar should appear. Select the "not installed" filter to see the libraries that have not been installed.
    e. Find Tensorflow and Keras libraries using the search feature
    f. Select the libraries and install
    g. To utilize these libraries in a Jupyter notebook, import them by entering this code:

        "import tensorflow as tf
        from tensorflow import keras"
4. Installing Numpy
    a. Anaconda comes preinstalled with Numpy so there is no need to install Numpy
    b. To import numpy into the jupyter notebook, write the following code:

        "import numpy as np"
5. Installing OpenCV
    a. First, open the Anaconda navigator that was previously installed
    b. Go to the environments tab on the left-hand side
    c. Click on the newly created environment from step 2
    d. On the right-hand side, a search bar should appear. Select the "not installed" filter to see the libraries that have not been installed.

e. Find OpenCV and Matplotlib libraries using the search feature

f. Select the libraries and install

g. To utilize these libraries in a Jupyter notebook, import them by entering this code:

“import cv2”

“import matplotlib.pyplot”

Procedure 2 - Creating the Convolutional Neural Network (CNN)

1. Setting up the workspace
   a. Open Anaconda
   b. Go to Jupyter Notebook via the newly created environment from procedure 1
   c. Create a new notebook to code
   d. Import the necessary libraries using the code from procedure 1 as well as the sequential model from tensorflow, all of the types of layers that will be used (see in step 2 of procedure 2), and finally import pickle.
2. Importing the data set and preparing it for usage
   a. Create 2 folders for brain tumor MRI’s: one for training and one for testing
   b. Load the data sets into the Jupyter notebook that is being used
      i. First, find the file location of the brain tumors.
      ii. Assign this file path a name.
      iii. Specify the categories that will be identified. In this case, it would be different tumor types.

iv. Next, the images need to be normalized and altered in a way that is read by the CNN in an optimal way.

v. For this project, the images will be resized to 150x150 and converted to grayscale.

   1. To accomplish this run the file path through a for loop with the changes. Use the function “img_array = cv2.imread(os.path.join(path,img), cv2.IMREAD_GRAYSCALE” and “new_array = cv2.resize(img_array, (150,150))”

vi. Finally, save this created data to pickle so it can be called for usage during the creation of CNN later on in the procedure.

c. Next, create classes for the types of tumor (classes = ["glioma_tumor", "meningioma_tumor", "no_tumor", "pituitary_tumor"].

d. Start a new Jupyter notebook for the creation of the CNN

e. Load the data using pickle using a function such as: “pickle.load(open("X.pickle","rb"))” and “pickle.load(open("y.pickle","rb"))”

f. Create a sequential model. These models are appropriate for a stack of layers where each layer has one input and output tensor. Use the following code:

   “cnn= models.Sequential([...])”

g. Add the first convolutional layer for feature extraction of the brain tumor. To do this, the number of filters used, the size of filters, the input shape, and the activation function must be specified. The number of filters determines how many features will be extracted. The filters do not have to be described by the programmer because the network will do it itself. To start, a random value such as 8 or 16 can be assigned for the number of filters. This number will later be tweaked based on the results and the accuracy of the program. The size of the filter that will be employed is 3x3. The activation function that will be used is relu. The input shape of the image is set to 150 pixels by 150 pixels.

h. The max pooling operation must be added for feature extraction as well. The max pooling layer will be kept at 2x2 and the activation function is relu.

i. The number of max pooling convolutional layers will vary depending on the required accuracy and project. To increase the number of convolutional and max pooling layers, repeat steps g and h.

j. A flattening layer will be added to make the multidimensional input into a one-dimensional input. The activation function for this is relu.

k. Dense layers will also be added to classify the image based on the outputs from the other layers and provide accuracy. The number of neurons in the dense layer must be specified. For convolutional neural networks, small amounts of neurons and actual dense layers are needed for success. This project will employ 2 dense layers. The one classifying the tumor will have 128 neurons and the relu activation function while the other one that will provide the accuracy will contain 64 neurons and a softmax activation function. Softmax will provide a normalized probability.

l. An optimizer must be implemented as well. The optimizer that will be used is “adam”. Sparse categorical cross entropy will be used alongside this.

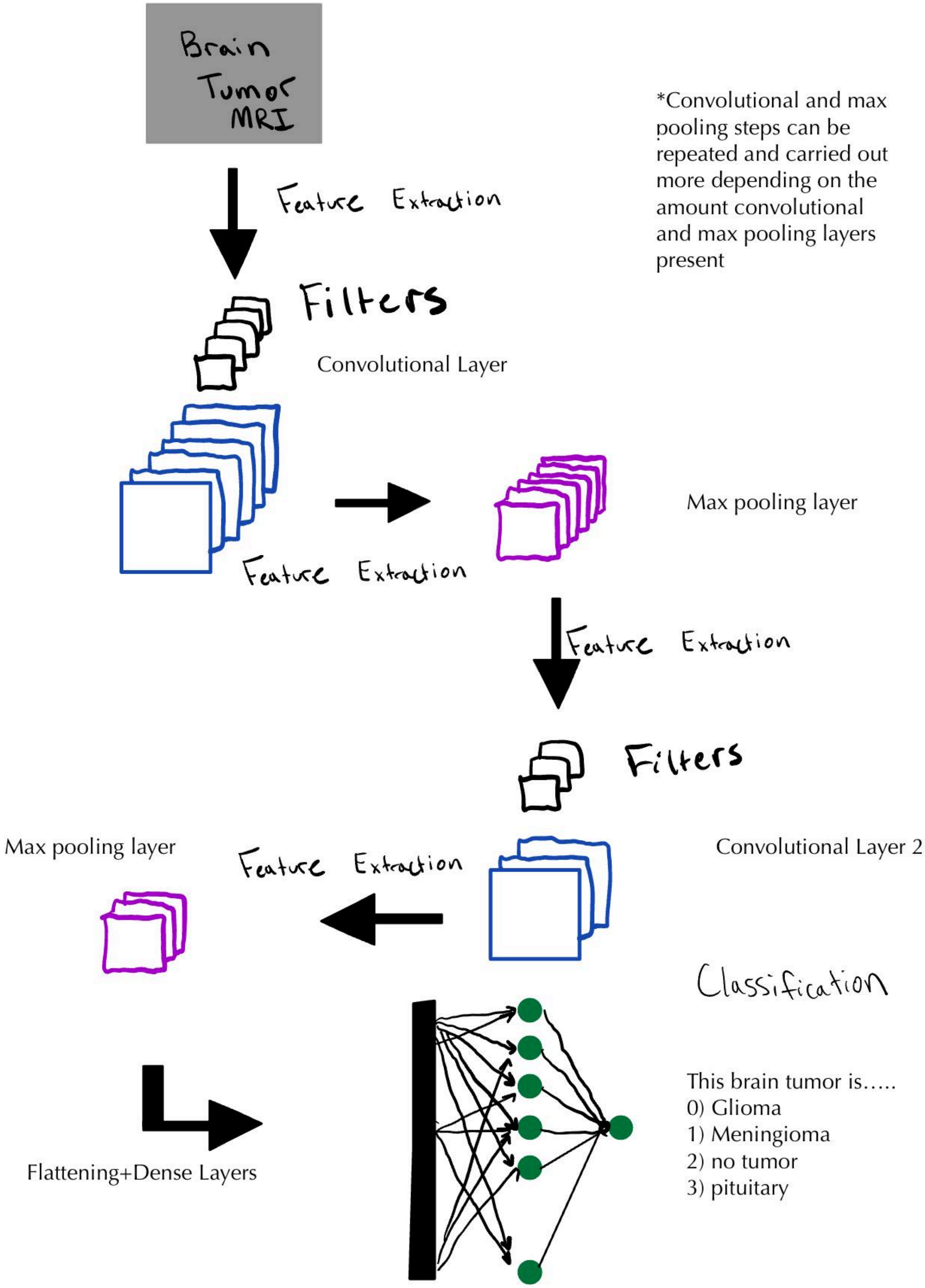
m. After the creation of the CNN ensure to save the model using the ".save" command.

n. To run the command specify the following: number of epochs, batch size, and validation split. The number of epochs tells the program how long to train for, in this case, there will be 20 epochs. Batch size specifies how many images will be looked at during training per epoch. Validation split is the percent of total images that will be used for testing to provide accuracy of programming.

Procedure 3: Training, Testing, and improving

After creating the neural network framework—using the training dataset— the program will be taught to recognize tumors. The fit function will run the program for a certain number of epochs (one epoch is the passing of a dataset through the neural network). This process will train the program. Then the program will be tested using the test data set via the evaluate function and the inputting of singular images. The accuracy of the program will be given after the program is tested. Based on the results, the number of convolutional layers, dense layers, max-pooling layers, and the characteristics of each will be altered using the outlined steps in part D of procedure 2. After this, training and testing will occur again and more changes will be made. This process will continue until the desired accuracy is reached. Under ideal conditions, an algorithm similar to this should reach 95 percent or higher, but given the constraints, the goal is to reach a minimum accuracy of 90 percent.

Experimental Set-up



*Convolutional and max pooling steps can be repeated and carried out more depending on the amount convolutional and max pooling layers present

This brain tumor is…..
0) Glioma
1) Meningioma
2) no tumor
3) pituitary

This diagram outlines the convolutional neural network used in this project visually. It shows all the layers that will be present and contribute to the analysis and classification of brain tumors.

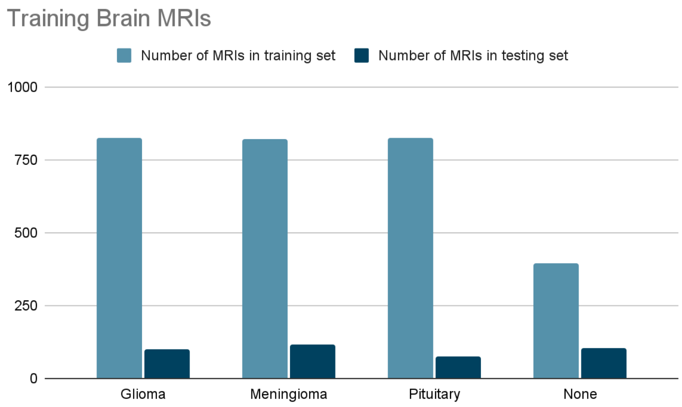
Data Collection

Trial	Description of CNN (number of convolutional layers, filters, etc.)	Percent accuracy (the number of correctly identified images divided by the total number of images)	Proposed method for improvement	T-test results
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				

The data that will be collected will illustrate the capabilities and overall performance of the convolutional neural network's ability to identify and classify brain tumors. Based on the results of the program, this table will display variations of the convolutional neural network design and the yielded accuracy. It will also display T-test results. T-tests show how significant the means are between group means. It will display if changes between accuracies obtained happened by chance or happened due to significant changes within a group and if that change was statistically significant. Bigger t-values equal a greater probability that the results are repeatable. With this information, the convolutional neural network can be modified and improved upon. This data will also help determine if artificial intelligence can be a viable alternative to traditional diagnosis methods performed by humans.

Data Analysis:

Figure 1a. Amount of Raw Training and Testing images



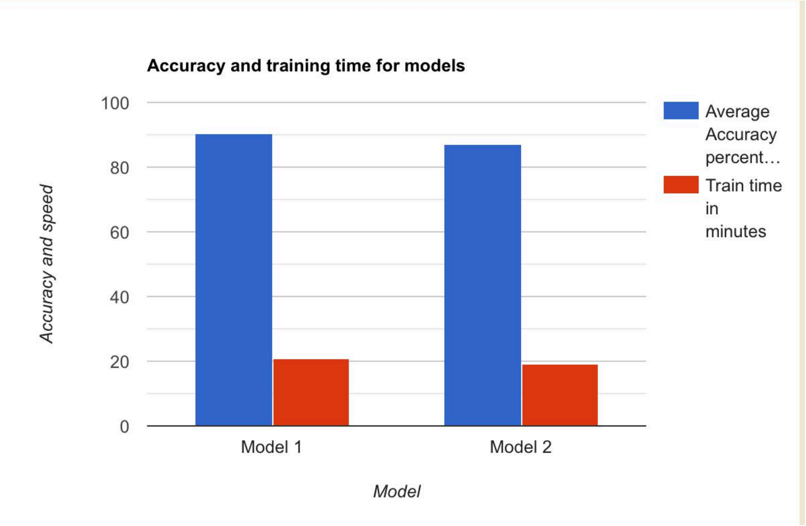
These graphs display the number of each type of brain tumor present within the testing and training datasets utilized. Figure 1a shows that the training dataset contains 826 glioma tumors, 822 meningioma tumors, 827 pituitary tumors, and 395 normal brain MRIs and includes 100 glioma tumors, 115 meningioma tumors, 74 pituitary tumors, and 105 normal brain MRIs in the testing dataset.

Table 1. Two Variations of CNN Models tested for the task

CNN	Convolution layers	Maxpooling layers	Dense Layers	Optimizer	Dropout layers	Validation accuracy	Average Speed for training
Model 1 (selected model for use in the project)	4	4	2	ADAM	5	Trial 1: 91.88 % Trial 2: 88.76% Trial 3: 90.93%	20 minutes 50 seconds
Model 2	3	3	2	SGD	4	Trial 1: 83.76 % Trial 2: 85.45% Trial 3: 87.78%	19 minutes 3 seconds

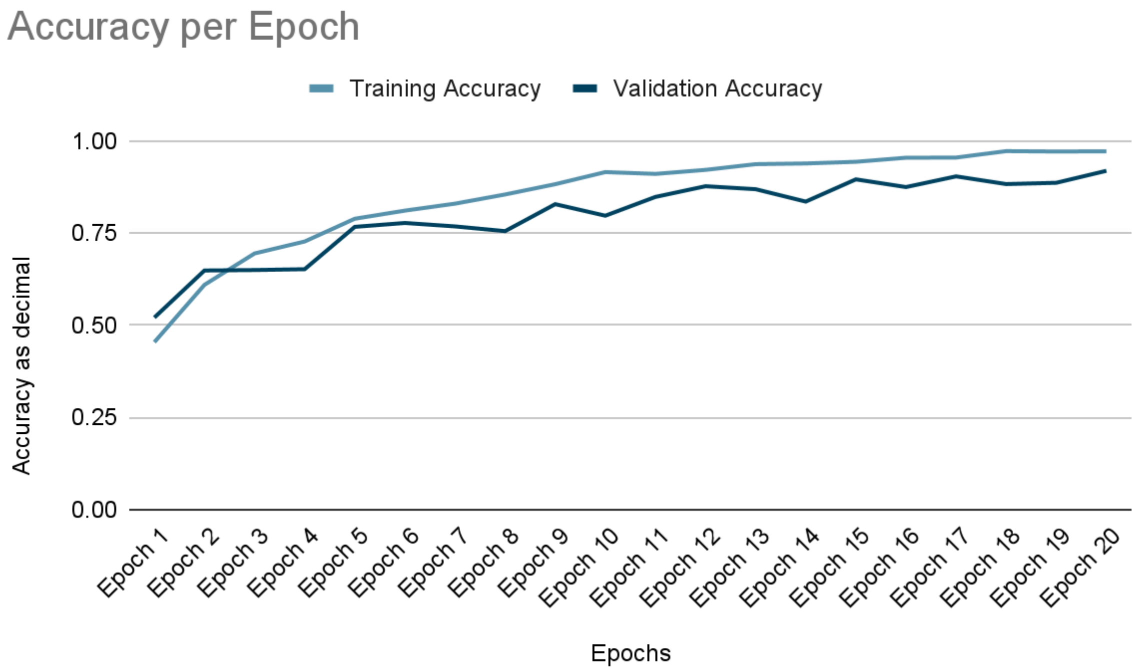
This table shows 2 different models that were used along with their characteristics. It also shows the validation accuracy and time taken for training the model for the project's specific use. Based on these results, the model to train and test the data was chosen.

Graph 1. Graph of Accuracy and Training time for each model



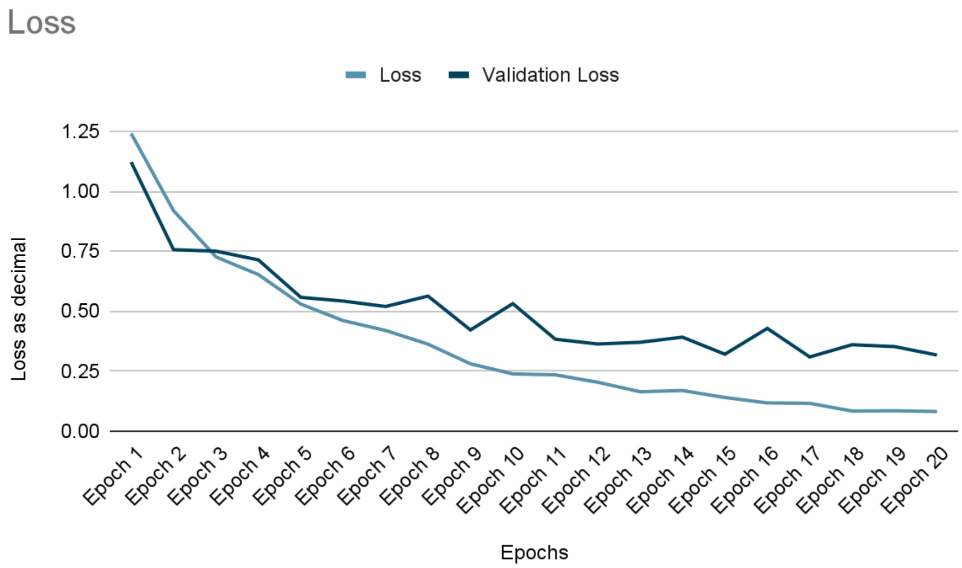
This graph displays the data outlined in Table 1. The blue graphs represent the average accuracy of the model and the red represents the train time in minutes. The average accuracy of model 1 was 90.52 percent while model 2 had an accuracy of 85.66 percent. After performing a t-test, 0.0327 was the obtained p-value.

Graph 2a. Training and Validation Accuracy of CNN model



This graph outlines both the accuracy of the model during training and the predicted accuracy on new data(validation accuracy) over the course of 20 epochs.

Graph 2b. Training Loss and Validation Loss of CNN model



This graph outlines both the loss of the model during training and the predicted loss of new data(validation loss) over the course of 20 epochs.

Table 2a. Results of CNN model during testing on new data (Trial 1)

Tumor Type	**Correct Diagnosis? (y/n)**	**Tumor Type**	**Correct Diagnosis? (y/n)**
Glioma	**y**	**Glioma**	**n**
Meningioma	**y**	**Meningioma**	**y**
Pituitary	**y**	**Pituitary**	**y**
None	**y**	**None**	**y**
Glioma	**y**	**Glioma**	**y**
Meningioma	**y**	**Meningioma**	**y**
Pituitary	**y**	**Pituitary**	**y**
None	**y**	**None**	**y**
Glioma	**y**	**Glioma**	**y**
Meningioma	**y**	**Meningioma**	**y**
Pituitary	**n**	**Pituitary**	**y**
None	**y**	**None**	**y**

This table displays what specific tumor the CNN analyzed and whether it correctly identified the tumor. The time for each diagnosis from the CNN model was one second. This data will be used to determine the accuracy of the model.

Table 2b. Accuracies after testing CNN

Type	**Accuracy**
Glioma	**83.33%**
Meningioma	**100%**
Pituitary	**83.33%**
None	**100%**
Total	**91.67 %**

This table shows the accuracy of the program during trial 1 as shown in Table 2a. The accuracy for each type of tumor is displayed, as well as the total accuracy of the program. These were calculated by taking the number of correct classifications and dividing them by the total number of classifications.

Table 3a. Accuracy over many trials

Trial number	Number of MRIs present	Total Accuracy
1	24	91.67%
2	24	91.67%
3	24	87.5%
4	24	95.83%
5	24	91.67%
6	24	87.5%
7	24	83.33%
8	24	87.5%
9	24	91.67%
10	24	87.5%

Average	89.584%

This table shows 10 trials of testing and their respective accuracies using the same process outlined in tables 2a and 2b. It also shows the average accuracy over all 10 trials. Each trial had a different set of 24 images.

Table 4a. Accuracy of specific tumor types

Tumor Type	Accuracy of program
Glioma	91.6%
Meningioma	91.2%
Pituitary	83.8%
None	92.4%

This table shows the program's accuracy on specific types of tumors. These averages were created using the data used to create table 3a.

As shown in Figure 1a, a dataset of 2870 brain MRI scans was utilized for training the CNN to identify and classify brain tumors. First, the model that makes decisions based on what it has learned had to be created. Two models were created and tested for this purpose. One utilized an Adam activation function while the other made use of stochastic gradient descent. The one that utilized the stochastic gradient descent had slightly fewer layers. These differences made an impact as the model that used Adam performed better and was therefore the model that was chosen for the project. The data for this is shown in table 1. The differences between the two models are also statistically significant. This was determined via a T-test and the P-values received from it. After executing the T-test, the P-value equaled 0.0327 as stated in graph 1. The standard criteria are to have a P-value less than 0.05 because that means the data is statistically significant and the null hypothesis is rare. Therefore, due to the p-value, the data is significant and there is an actual difference between both models.

Next, the program was trained for the usage of classification. There are a few guidelines for training to ensure that everything is being trained properly and is in the best method. This can be determined by graphs 2a and 2b. Graph 2a represents the validation accuracy and training accuracy while Graph 2b shows the validation loss and training loss. When the validation accuracy— the predicted accuracy of new data— is close to the training accuracy, which is the accuracy of a familiar dataset, it means that the model fits the data well. If there is a significant decrease in validation accuracy, it can be concluded that the model is overfitting. This means that the model is too familiar with the training data and is having difficulty interpreting new samples. For both validation loss and training loss, the lower they are, the better. This is a metric to see the number of errors present during the training and testing phases and these values should also be close together. Since the validation and training accuracies are both high and close to each other and the loss is low, it can be concluded that this model properly fits its purpose and is able to classify new and unfamiliar data.

Additionally, the data has to be analyzed with new brain tumor MRI samples to see how the model actually performs and its accuracy. For this test, 24 images were chosen: 6 glioma tumors, 6 meningioma tumors, 6 pituitary tumors, and 6 normal brain scans. After this, these images were run through the CNN model, and the classification was recorded in Table 2a. Each prediction of what brain tumor was present took around one second. In the first trial (Table 2a), out of all samples, one pituitary and one glioma tumor were misidentified. This results in an 83.33 percent accuracy on glioma tumors, 100 percent with meningioma tumors, 83.33 percent with pituitary tumors, and 100 percent with normal tumors. This resulted in an overall accuracy of 91.67 percent. These conclusions are presented in Table 2b. This mirrors the predicted accuracies outlined in Table 1. The process outlined for testing the brain tumors was repeated 9 more times with different sets of images and the results are shown in Table 3a. The data from Table 3a shows that this program is fairly consistent with its classification of different types of brain tumors due to the fact that the total accuracy percentage over the 10 trials was 89.584 percent which is close to 90 percent. There were a few outliers where the program performed worse or even better, but in general, the accuracy stayed consistent at around 90 percent.

Finally, by analyzing data collected from table 3a, the accuracy of the program on each type of tumor can be derived. The derived accuracies are shown in table 4a. Glioma tumors were correctly identified 91.6 percent of the time, meningioma was correctly identified 91.2 percent of the time, the pituitary tumors were correctly identified 83.8 percent of the time, and finally, the no tumor images were correctly identified 92.4 percent of the time. From this data, it can be deduced that the program struggled the most with pituitary tumor recognition. This error could be due to how similar pituitary tumors may look to other tumors from the perspective of the program. More data could be a possible way to combat this as it will then have more information to use to learn distinctions between the types of tumors.

Data Conclusion:

Every second could mean the difference between life and death for a patient when it comes down to diagnosing and classifying brain tumors. Now, more than ever, is a time when it is imperative to develop better ways to diagnose life-threatening complications like brain tumors with the use of new technology. Delays and misinterpretations of data will slow the speed of the administration of treatment, which could have disastrous consequences for a patient, making it all the more important to reach conclusions as fast and accurately as possible. Currently, brain tumors are being classified by radiologists by performing an MRI. In 4 percent of cases, brain tumors can be misidentified or overlooked, meaning radiologists, in general, are 96 percent accurate. This may seem like a very small amount of error, but when considering that there are hundreds of thousands of people who test for brain tumors, it is evident that a lot of lives are at stake. The factors of speed and accuracy of current methods motivate and entertain the ideas of creating new methods. Artificial intelligence, in particular, convolutional neural networks can pave the way for an automated method of diagnosis. These models have been proven to be effective, accurate, and fast. The goal of this experiment was to create a convolutional neural network that has the ability to identify different brain tumors accurately and quickly. The model that was created in this project was less than 96 percent accurate which makes it inferior to current methods. However, this project succeeded in providing a possible method that when refined, can eventually become a viable method for quick and accurate diagnosis.

To begin this experiment, a computer and access to the internet were required. The recommended specifications of the computer for the best experience had 16 gigabytes of RAM, at least 256 gigabytes of storage, and a fairly new CPU (i.e. 8th generation intel CPU and onwards). The applications to run and code this program, such as anaconda and Jupyter notebook, are also required. Next, the dataset of brain tumors needs to be imported and made usable by convolutional neural networks. The dataset used in this project is from Kaggle. The images have to be converted into arrays, iterated, converted to grayscale, and then resized. After this, the images are ready to be used by the program.

The next step of the experiment is to establish the CNN model by defining it and adding the convolutional layers, max-pooling layers, dense layers, flattening layers, and dropout layers. Many models can be created for this type of project. In this project, two were created and tested. After data was collected, a t-test was performed and a p-value was extracted to ensure that the differences between these models were actually significant. Next, after the most advantageous model has been chosen, the model needs to be compiled before fitting it and starting the training process. The compile function contains the Adam optimizer and uses sparse categorical cross entropy for the loss values. To begin the training of this program the model fit function will be used. It will contain the batch size, the number of epochs the program will run for as well as the validation split. After the training finishes the final step is to test it with new data: in this case, new MRI scans. Data from both the training and testing phases will be collected.

After training the program with the model that was chosen, the predicted accuracy (validation accuracy) was 90.52 percent. From the ten trials of the testing procedure, it can be seen that the accuracy was mostly hovering between 87.5 to 91.67 percent accurate with a few outliers where the accuracy was significantly different from the rest. This shows how this model was both accurate and consistent. Each prediction only took around one second, meaning that this model is also very fast. A possibility for why there were trials where the accuracy was less than the predicted accuracy within the training process is due to overfitting. Minor overfitting within this model could have made this model too fit for its dataset and made it more difficult for it to classify images outside of the training dataset. This possibility could be fixed with changes to the layers within the CNN model or training for a different number of epochs. Although the speed and accuracy of this program are accurate and fast, it is not on par with the accuracy of humans and can be improved significantly.

Due to the lack of time and vast amounts of data, the CNN model could not perform at a higher level and would have errors. Without more data, it was more difficult for the model to be exposed to more outcomes of brain tumors. Therefore when faced with possibilities that it may have not seen before, it had a more difficult time recognizing what type of tumor was present.

This presented itself as a lower accuracy percentage. The time constraint on this project also made it so there was less time to optimize this program, as well as less time to run the program. Running larger models such as this one for brain tumors means that training the model after each change to analyze the data takes a lot of time. With more time this model could be further improved by increasing accuracy and can even be given the ability to identify specific types and subcategories of each tumor.

The primary objective of this project was to use artificial intelligence, in particular convolutional neural networks within deep learning, to create a model that can classify brain tumors quickly and with a minimum of 90 percent accuracy. In an ideal world, the STEM and medical community will look at this code to normalize the path to the integration of artificial intelligence within more environments such as hospitals. It will provide a template for other image recognition tasks as well. If artificial intelligence continues to show evidence that it is viable to use, it will bring forth lots of benefits. Diagnosis becomes faster, as well as removing human error. This will save countless lives due to earlier diagnosis, allowing for more time for treatment. These programs, if trained accordingly, can also be more accurate than radiologists, saving even more lives. If not trusted with full diagnosis, at the very minimum, programs such as this can be used as a safeguard or as a second set of eyes for diagnosis. Radiologists can reach their own conclusions about diagnosis and then use artificial intelligence to verify their findings. Based on the results of this project, two questions arise: when can artificial intelligence be clinically accepted as a method to diagnose patients, or at minimum, as an aid to radiologists? What other real-world applications can similar CNN models be used for?

Works Cited

Aboutcancer.com. (n.d.). MRI Appearance of Primary Brian Tumors. MRI picture of a glioblastoma. Retrieved October 1, 2022, from http://www.aboutcancer.com/mri_gbm.htm

Aggarwal, C. C. (2019). Neural networks and deep learning: A textbook. Springer.

Brady, A. P. (2017, February). Error and discrepancy in radiology: Inevitable or avoidable? Insights into imaging. Retrieved September 26, 2022, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5265198/

Bressler, N. (2021, November 10). How to Check the Accuracy of Your Machine Learning Model. Deepchecks. Retrieved October 15, 2022, from https://deepchecks.com/how-to-check-the-accuracy-of-your-machine-learning-model/

Cafasso, J. (2018, June 9). Proto-oncogenes: Definition, function, and relation to cancer. Healthline. Retrieved October 1, 2022, from https://www.healthline.com/health/proto-oncogene#cancer

Cancer Treatment Centers of America. (2022, February 28). Oncogenes. Cancer Treatment Centers of America. Retrieved September 27, 2022, from https://www.cancercenter.com/risk-factors/oncogenes

Cancer.net. (2021, November 10). Brain tumor - diagnosis. Cancer.Net. Retrieved October 1, 2022, from https://www.cancer.net/cancer-types/brain-tumor/diagnosis

Cleveland Clinic. (2022, June 2). Brain Tumor: Symptoms, Signs & Causes. Cleveland Clinic. Retrieved October 14, 2022, from https://my.clevelandclinic.org/health/diseases/6149-brain-cancer-brain-tumor

Cornelisse, D. (2018, February 26). An intuitive guide to Convolutional Neural Networks. freeCodeCamp.org. Retrieved October 1, 2022, from https://www.freecodecamp.org/news/an-intuitive-guide-to-convolutional-neural-networks-260c2de0a050/

Dana-Farber Institute. (2022). Brain tumors. Dana-Farber. Retrieved October 1, 2022, from https://www.dana-farber.org/brain-tumors/diagnosis/

Domino. (n.d.). What is Anaconda? | Domino Data Science Dictionary. Domino Data Lab. Retrieved October 16, 2022, from https://www.dominodatalab.com/data-science-dictionary/anaconda

Driscoll, M. (2022, July 28). Jupyter Notebook: An introduction. Real Python. Retrieved October 1, 2022, from https://realpython.com/jupyter-notebook-introduction/

Fahrenhorst-Jones, T. (2022, April 28). Meningioma: Radiology reference article. Radiopaedia Blog RSS. Retrieved October 1, 2022, from https://radiopaedia.org/articles/meningioma?lang=us

Gupta, A. (2022, August 22). Some amazing applications of opencv library. Analytics Vidhya. Retrieved October 1, 2022, from https://www.analyticsvidhya.com/blog/2021/09/some-amazing-applications-of-opencv-library/

John Hopkins Medicine. (2021, November 8). Brain tumor types. Johns Hopkins Medicine. Retrieved October 1, 2022, from https://www.hopkinsmedicine.org/health/conditions-and-diseases/brain-tumor/brain-tumor-types

John Hopkins Medicine. (2022, February 11). Gliomas. Johns Hopkins Medicine. Retrieved October 1, 2022, from https://www.hopkinsmedicine.org/health/conditions-and-diseases/gliomas

John Hopkins Medicine. (2021, November 4). Meningioma. Johns Hopkins Medicine. Retrieved October 1, 2022, from https://www.hopkinsmedicine.org/health/conditions-and-diseases/meningioma

John Hopkins Medicine. (2021, August 8). Pituitary tumors. Johns Hopkins Medicine. Retrieved October 1, 2022, from https://www.hopkinsmedicine.org/health/conditions-and-diseases/pituitary-tumors

Kadlaskar, A. (2021, June 14). Image classification using CNN: Python implementation. Analytics Vidhya. Retrieved October 1, 2022, from https://www.analyticsvidhya.com/blog/2021/06/image-classification-using-convolutional-neural-network-with-python/

Khan, A. N. (2022, October 5). Pituitary adenoma imaging: Practice Essentials, computed tomography, magnetic resonance imaging. Retrieved October 1, 2022, from https://emedicine.medscape.com/article/343207-overview

Lang, N. (2022, October 1). Using convolutional neural network for Image Classification. Medium. Retrieved October 3, 2022, from https://towardsdatascience.com/using-convolutional-neural-network-for-image-classification-5997bfd0ede4

Lynne Eldridge, M. D. (2022, June 6). What are oncogenes and how do they cause cancer? Verywell Health. Retrieved September 29, 2022, from https://www.verywellhealth.com/oncogenes-types-and-role-in-cancer-4178292

Mayo Foundation for Medical Education and Research. (2021, August 6). Brain Tumor. Mayo Clinic. Retrieved September 27, 2022, from https://www.mayoclinic.org/diseases-conditions/brain-tumor/symptoms-causes/syc-20350084

National Human Genome Research Institute. (2022, October 7). Apoptosis. Genome.gov. Retrieved October 1, 2022, from https://www.genome.gov/genetics-glossary/apoptosis

NCI Staff. (2020, February 12). Artificial Intelligence Expedites Brain Tumor Diagnosis. National Cancer Institute. Retrieved September 25, 2022, from https://www.cancer.gov/news-events/cancer-currents-blog/2020/artificial-intelligence-brain-tumor-diagnosis-surgery

Onder, O., Yarasir, Y., Azizova, A., Durhan, G., Onur, M. R., &amp; Ariyurek, O. M. (2021, April 20). Errors, discrepancies and underlying bias in radiology with case examples: A pictorial review - insights into imaging. SpringerOpen. Retrieved September 27, 2022, from https://insightsimaging.springeropen.com/articles/10.1186/s13244-021-00986-8

Padmanaban, S. (2020, January). MRI planes for MRI head scan (A) axial (b) coronal (c) sagittal mr ... Research Gate. Retrieved October 9, 2022, from https://www.researchgate.net/figure/MRI-planes-for-MRI-head-scan-a-Axial-b-Coronal-c-Sagittal-MR-scanner-can-generate_fig2_338448026

Simplilearn. (2022, July 24). What is TensorFlow? Deep Learning Libraries and program elements explained. Simplilearn.com. Retrieved October 1, 2022, from https://www.simplilearn.com/tutorials/deep-learning-tutorial/what-is-tensorflow

Stempniak, M. (2022, March 18). Common reasons radiologists missed head and neck cancer diagnoses on cross-sectional imaging. Radiology Business. Retrieved September 26, 2022, from https://radiologybusiness.com/topics/care-delivery/healthcare-quality/radiologist-missed-head-and-neck-cancer-imaging

Wolfewicz, A. (2022, October 5). Deep Learning vs. machine learning – what's the difference? Levity. Retrieved October 7, 2022, from https://levity.ai/blog/difference-machine-learning-deep-learning

W3Schools.com. (n.d.). Introduction to NumPy. W3Schools. Retrieved October 16, 2022, from https://www.w3schools.com/python/numpy/numpy_intro.asp

Yadav, S. S., & Jadhav, S. M. (2019). Deep convolutional neural network-based medical image classification for disease diagnosis. *Journal of Big Data*, *6*(1), 1-18. https://doi.org/10.1186/s40537-019-0276-2