Brain Tumor Classification: Using CNN's to help diagnose and label brain tumors

Cordelia Sherwood Follow 7 min read · Dec 9, 2024

By Cordelia Sherwood, Rahil Verma, Spencer Katz & Al Intisar

Introduction/Overview

In this project we used convolutional neural networks to classify brain tumors based on a dataset with around 7,000 MRI's with four different classes: glioma ,meningioma , pituitary , and no tumor. According to *A Review of Recent Advances in Brain Tumor Diagnosis Based on AI-Based Classification,* "brain scans or image interpretation to diagnose illnesses are prone to inter-reader variability and accuracy, which depends on the medical practitioner's competency." A machine learning model that can classify the tumor types with high accuracy can reduce misdiagnoses and major upheaval and stress in people's lives.

You can find our code and other information here: ***GitHub Repository***

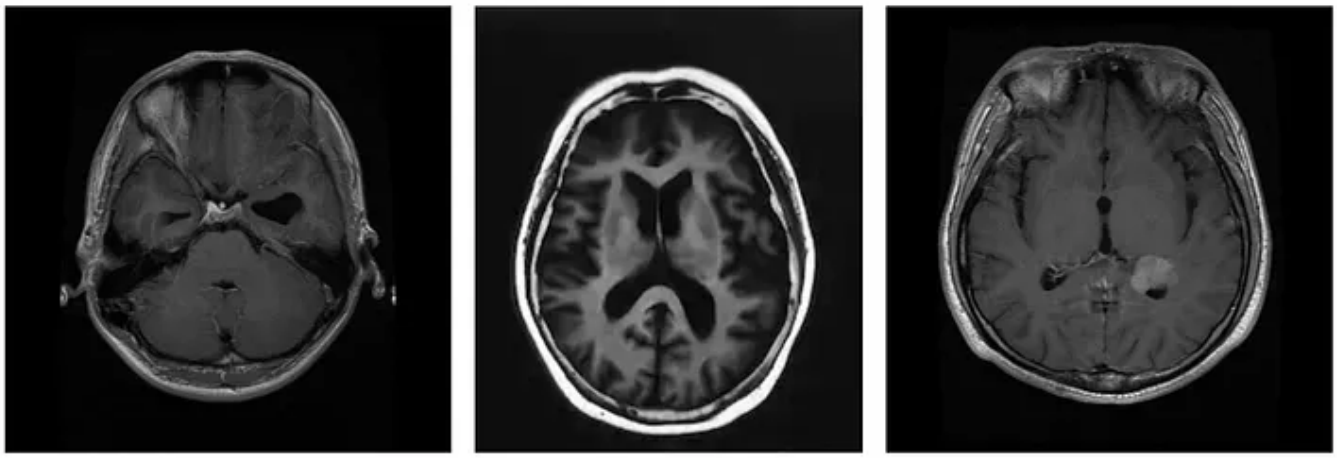


Figure 1: Brain tumor MRI images.

Methodology & Data

Our data was sourced through Kaggle which contains a downloadable dataset that contained about 7,000 images over the four classes. We processed them into testing and training folders by using scikit-learn to split the data. One thing to note — the images come from 7 different datasets, comprising of various angles and cross sections. This introduces confounding variables such as orientation that the model might learn. For our model, we preprocessed the brain tumor MRI images to size 120x120px using padding. Then we trained the model to classify the 4 different types of tumor images types.

Dataset Breakdown:

Label	#
Glioma Images	1600
Meningioma Images	1405
Pituitary Images	1316
No Tumor Images	1296

Figure 2: Training Data Breakdown

Label	#
Glioma Images	400
Meningioma Images	352
Pituitary Images	329
No Tumor Images	325

Figure 3: Testing Data Breakdown

Training the Dataset to Classify Brain Tumors in MRIs

We used a CNN (Convolutional Neural Network) to classify brain tumors. CNN's are effective for image classification because they can learn spatial hierarchies of features from input images. It does this by detecting patterns, like edges and textures, in visual data by using layers of filters (convolutions).

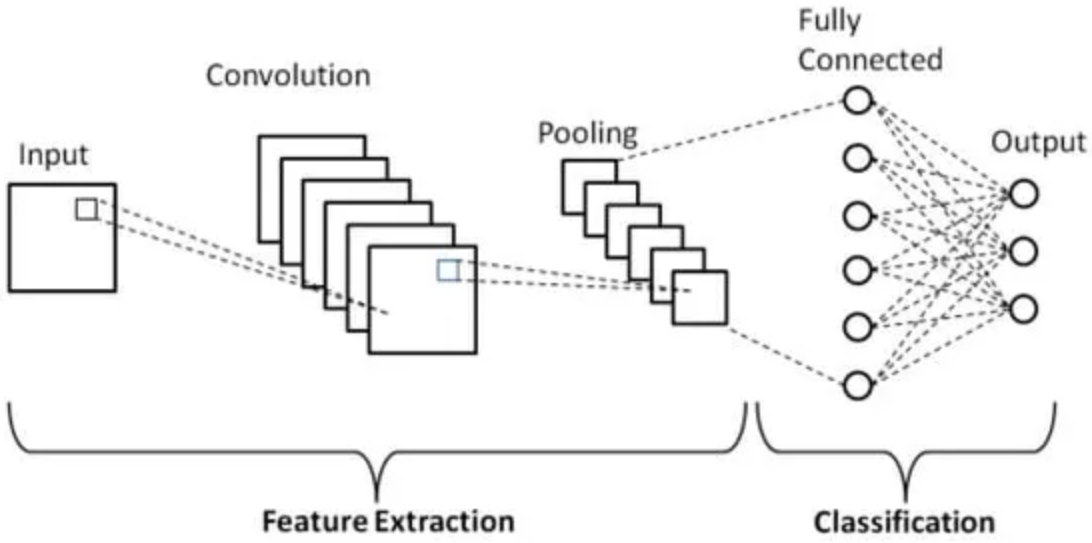


Figure 4: CNN Model Example

The CNN model architecture consists of input dimensions (120, 120, 3), which then goes through a sequence of convolutional pooling layers that detects low-level features and outputs a vector of length 4. The Pooling layers reduces spatial dimensions while retaining the important data (e.g. MaxPooling). And the additional convolutional layers continue to move into more complex patterns. Before the output layer, we used Global Average Pooling (GAP). GAP is often preferred over flattening in CNN architectures because it reduces the number of parameters, improves robustness, and minimizes the risk of overfitting. The comparatively smaller 1D vector produced by GAP allowed for faster training than the flatten layer by reducing the trainable weights in the subsequent fully connected layer. Our

final output will come in the form of probabilities for each of the four classes using SoftMax activation.

> ***Side Note:*** *One of the challenges that we faced initially was how our model was being fed the images. The encoding from original image to number resulted in too large vectors and crashed while training due to limited RAM. Also initially our model architecture was too complex with too many trainable parameters. Our model was too large, so we reduced the number of convolutional layers. Currently our model has 2308 trainable parameters.*

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 118, 118, 4)	112
batch_normalization (BatchNormalization)	(None, 118, 118, 4)	16
max_pooling2d (MaxPooling2D)	(None, 59, 59, 4)	0
dropout (Dropout)	(None, 59, 59, 4)	0
conv2d_1 (Conv2D)	(None, 57, 57, 8)	296
batch_normalization_1 (BatchNormalization)	(None, 57, 57, 8)	32
max_pooling2d_1 (MaxPooling2D)	(None, 28, 28, 8)	0
dropout_1 (Dropout)	(None, 28, 28, 8)	0
conv2d_2 (Conv2D)	(None, 26, 26, 16)	1,168
batch_normalization_2 (BatchNormalization)	(None, 26, 26, 16)	64
max_pooling2d_2 (MaxPooling2D)	(None, 13, 13, 16)	0
dropout_2 (Dropout)	(None, 13, 13, 16)	0
global_average_pooling2d (GlobalAveragePooling2D)	(None, 16)	0
dense (Dense)	(None, 32)	544
dropout_3 (Dropout)	(None, 32)	0
dense_1 (Dense)	(None, 4)	132

Total params: 2,364 (9.23 KB)
Trainable params: 2,308 (9.02 KB)
Non-trainable params: 56 (224.00 B)

Figure 5: Model Summary

Figure 5 is a summary of the model's architecture, including details about its layers, the number of parameters, and the overall structure. The figure highlights the types of layers used (e.g., convolutional, dense, dropout) along

with the input and output shapes at each layer, and the number of parameters that are trainable and non-trainable.

Evaluation

We used test loss and accuracy for evaluating the performance our model. Test loss measures how well the model's predicted probability distributions align with the actual class labels, calculated using the same loss function as during training (e.g., categorical cross entropy). It reflects the average "error" across all test samples, with lower values indicating better performance. Test accuracy, on the other hand, represents the proportion of test samples that the model classifies correctly by selecting the class with the highest predicted probability. Low loss and high accuracy suggest the model is both confident and correct, while discrepancies between them might highlight overconfidence or issues like overfitting.

Figures 6–9 show the evaluation of the model through graphs and summaries.

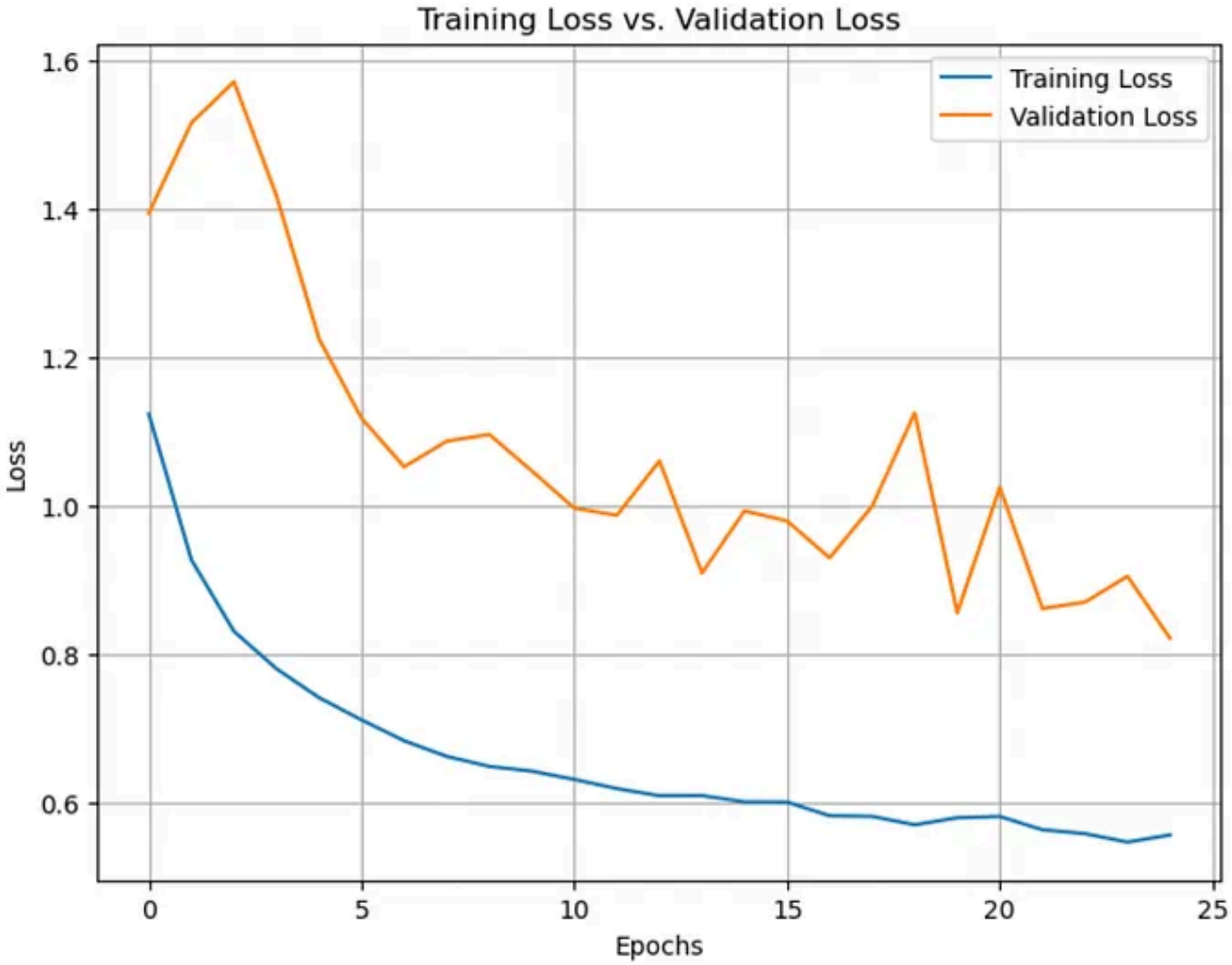


Figure 6: Training Loss vs. Validation Loss

Figure 6 compares the training loss and validation loss over the course of the training epochs. The plot shows how the model's training loss decreased steadily for all the 25 epochs but the validation loss had inconsistent ups and downs after the 10th epoch. This suggests the model might have overfitted after the 10th epoch.

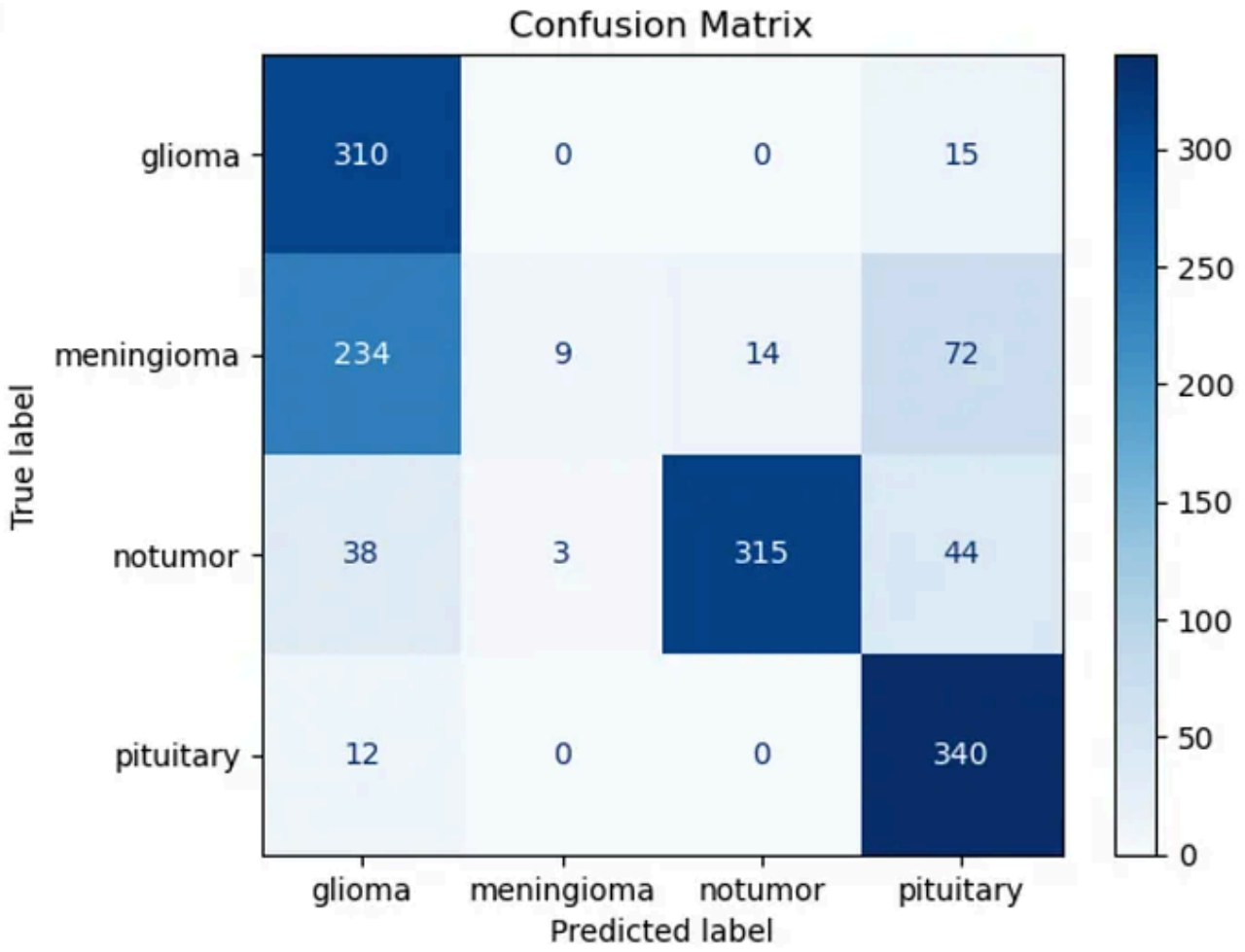


Figure 7: Confusion Matrix

Figure 7 displays the confusion matrix for a multi-class classification model. Each row of the matrix corresponds to the true class labels, while each column represents the predicted class labels. The diagonal cells show the number of correctly classified instances for each class, while off-diagonal cells indicate misclassifications, with their values reflecting the frequency of specific errors. As we can see the classes glioma, notumor, and pituitary were classified mostly correctly in the testing dataset but the class meningioma was misclassified the most as glioma. This indicates the model is not good at differentiating between glioma and meningioma tumors.

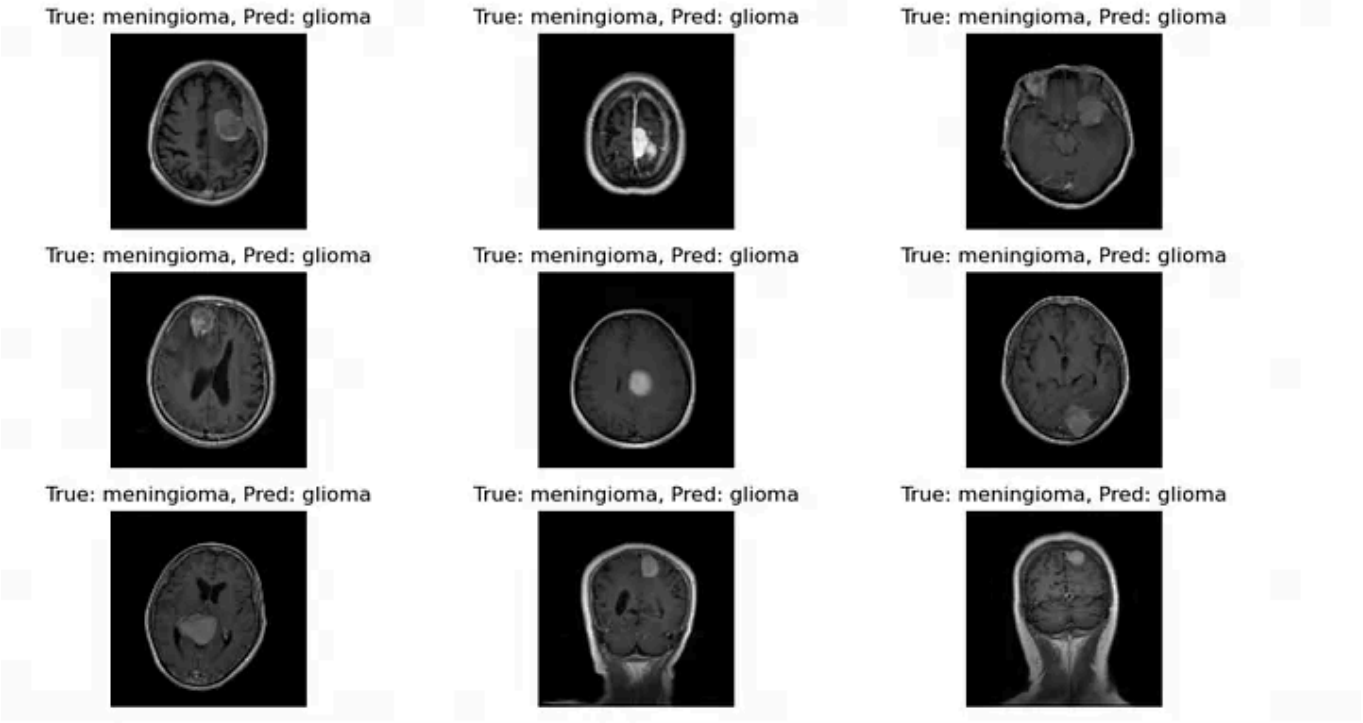


Figure 8: Misclassified Images

Figure 8 visualizes the misclassified images that were incorrectly classified as "glioma" when they actually belong to the "meningioma" class. After looking at some of the glioma images as well we saw the orientations of the glioma and meningioma were similar which might have resulted in the model to misclassify them. Overall, the model seems to able to differentiate between tumor and non-tumor pretty well. Additionally, the pituitary tumors are located in a specific location in the brain as opposed to glioma and meningioma which are not specific to a brain location.

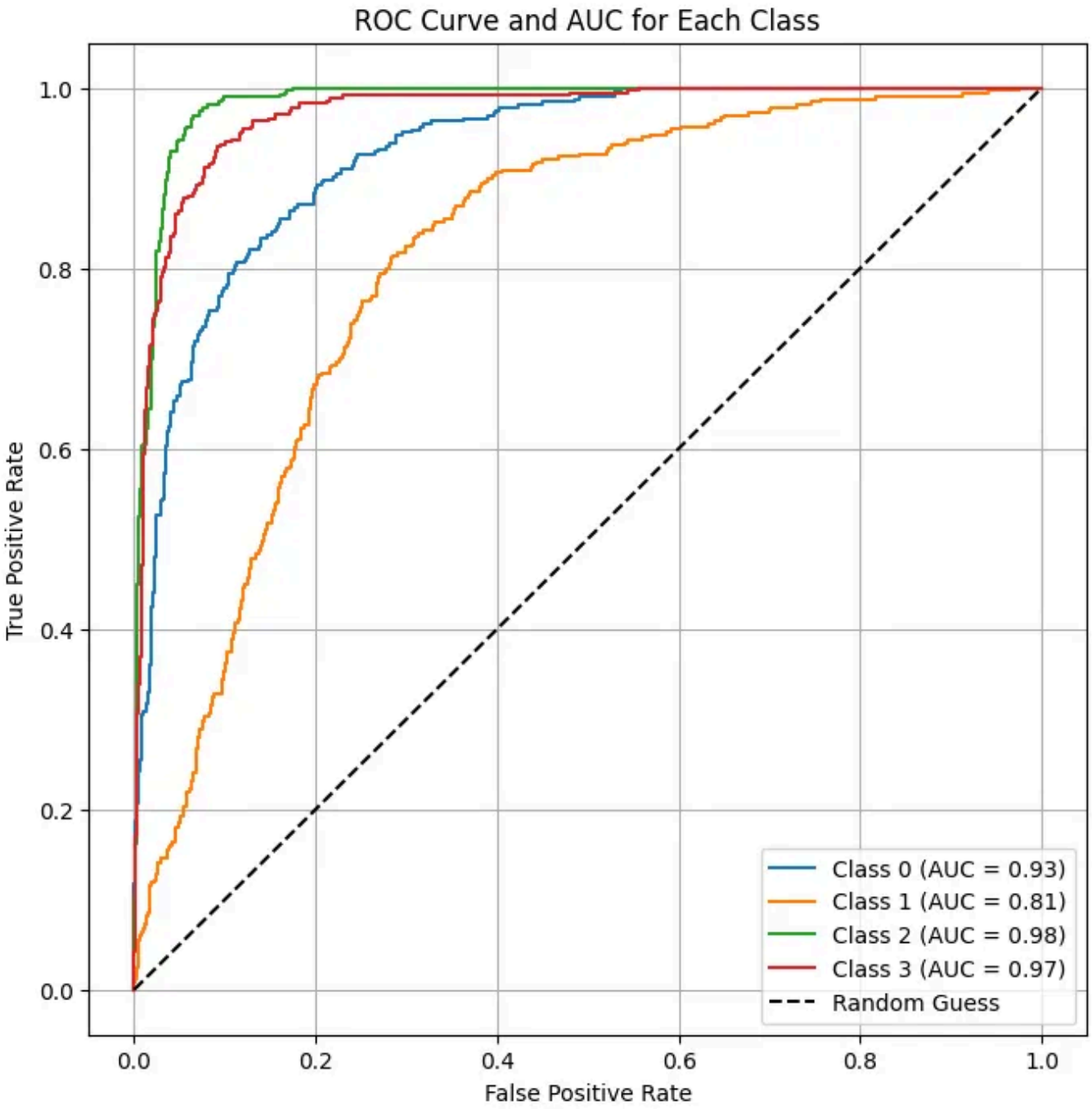


Figure 9:

Figure 9 visualizes Receiver Operating Characteristic (ROC) curves for a multi-class classification model, illustrating its performance for each class (classes: 0 = glioma, 1 = meningioma, 2 = no tumor, and 3 = pituitary). The ROC curve plots the True Positive Rate (TPR) against the False Positive Rate (FPR) across different classification thresholds, showcasing the model's ability to distinguish between classes. Each curve represents one class, and

the Area Under the Curve (AUC) value, displayed in the legend, quantifies the model's performance for that class: a higher AUC indicates better discrimination. The diagonal line represents a random guess, serving as a baseline for comparison. This figure reinstates our findings that the model is not good at classifying meningioma tumors.

Discussion

Our CNN model for brain tumor classification achieved moderate success with an accuracy of **69.46%**, effectively distinguishing between tumors and non-tumors and excelling in identifying pituitary tumors. However, challenges remain, particularly in differentiating gliomas from meningiomas due to overlapping features and limited computing powers for training. The model also showed signs of overfitting after the 10th epoch, highlighting the need for improved model architecture in terms of regularization and training techniques. Future enhancements could include transfer learning with larger pre-trained models, and the use of higher-resolution images. These improvements, combined with clinical validation from health professionals, could significantly enhance the model's reliability and pave the way for its real-world application in aiding accurate brain tumor diagnoses.

Conclusion

The prospect of a accurate machine learning model able to classify brain tumor types can immensely affect the efficiency and accuracy of brain tumor diagnosis. In light of this, we created a CNN that classifies brain tumors with 69.46 % test accuracy for MRI images of size 120x120px. By automating the analysis of medical imaging, CNNs can assist radiologists and doctors in detecting and classifying tumors like gliomas, meningiomas,

and pituitary tumors with high precision, potentially reducing diagnostic errors and improving patient outcomes. As the technology evolves, CNNs hold the promise of transforming medical diagnostics and advancing personalized medicine. Currently our model has test accuracy of about 69%, however a bigger model created with more computing power could become far more trustworthy in classifying brain tumors from brain MRIs.

Sources:

Phung, V.H.; Rhee, E.J. A High-Accuracy Model Average Ensemble of Convolutional Neural Networks for Classification of Cloud Image Patches on Small Datasets. Appl. Sci. ***2019****,* 9*, 4500.* https://doi.org/10.3390/app9214500

Nickparvar, Masoud. Brain Tumor MRI Dataset. Kaggle, 2021, https://www.kaggle.com/datasets/masoudnickparvar/brain-tumor-mri-dataset/data*.*

"Statistics and Research on Cancer Misdiagnosis in the NHS." Medical Negligence Assist, 2024, https://www.medicalnegligenceassist.co.uk/faqs/what-is-the-average-settlement-for-cancer-misdiagnosis/research-statistics-cancer-misdiagnosis-and-delayed-diagnosis-nhs*.*

Kaifi, Reham. "A Review of Recent Advances in Brain Tumor Diagnosis Based on AI-Based Classification." Diagnostics (Basel, Switzerland) *vol. 13,18 3007. 20 Sep. 2023, doi:10.3390/diagnostics13183007*
https://medium.com/@msong507/understanding-the-roc-auc-curve-cc204f0b3441

Written by Cordelia Sherwood

0 followers · 1 following

Follow

No responses yet

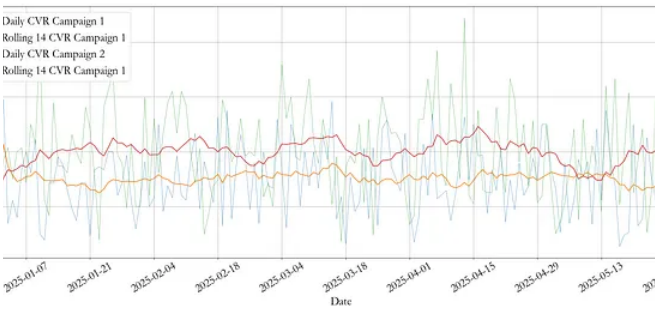
Write a response

What are your thoughts?

Recommended from Medium

In Long. Sweet. Valuable. by Ossai Chinedum

I'll Instantly Know You Used Chat Gpt If I See This



Danish Wang

Data Analysis Case Study: Campaign Comparison and...

Trust me you're not as slick as you think

May 16 23K 1397

Chapter I: Background

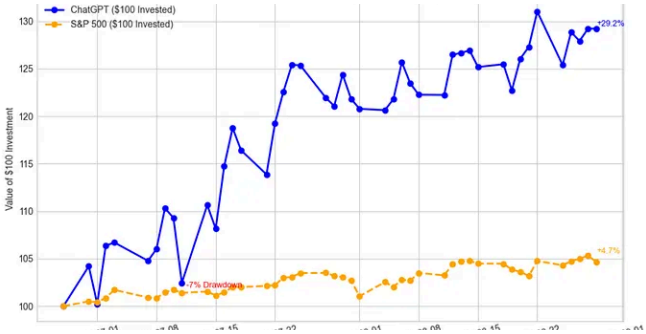
Sep 11

In How To Profit AI by Mohamed Bakry

Your ChatGPT History Just Went Public on Google. Here's What I Di...

Safety/Privacy Check Prompt Template Is Included

2d ago 9.3K 280

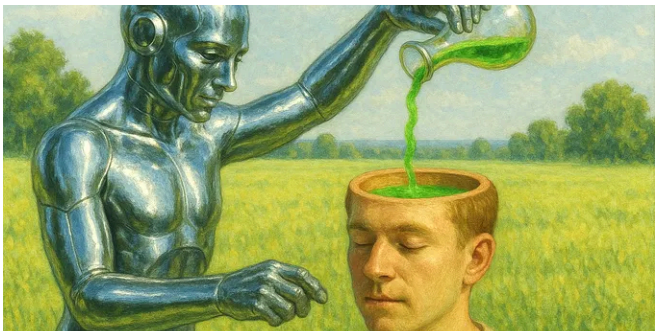


In Coding Nexus by Civil Learning

I Handed ChatGPT $100 to Trade Stocks—Here's What Happened i...

What happens when you let a chatbot play Wall Street? It's up 29% while the S&P 500...

Sep 2 2.7K 68



Jordan Gibbs

ChatGPT Is Poisoning Your Brain...

Here's How to Stop It Before It's Too Late.

Apr 29 25K 1281

In 3Streams by Cam Silver, Ph.D.

Teaching Charlie Kirk to My Students

Satire on August 27, Death on September 10

Sep 11 9.1K 240

See more recommendations