Classifying Brain Tumors from MRI Images With Image Discrepancies

Suzanna R. Thompson[*]
Interdisciplinary Data Science
Duke University
Durham, NC 27707
srt32@duke.edu

Abstract

The aim of this project is to develop a machine learning model that can accurately classify three types of brain tumors (glioma, meningioma, and pituitary) and "no tumor" using MRI images. The study investigates the impact of various types of image noise and resolution changes on the classification accuracy. Gaussian, Poisson, and quantization noise are added to the images, and the resolution is altered to simulate imaging artifacts. A convolutional neural network (CNN) is trained on the noisy images and evaluated on a test set to determine its accuracy in distinguishing between the different tumor types and healthy tissue. The results of this study may aid in the development of more accurate and robust classification models for brain tumors using MRI imaging data.

1 Introduction

Brain tumors are a major public health concern, with high mortality and morbidity rates worldwide. According to the American Brain Tumor Association, primary brain tumors account for 15% of all cancer diagnoses, and the incidence rate of malignant brain tumors has been increasing in recent years. Accurate diagnosis and classification of brain tumors are essential for proper treatment planning and improving patient outcomes.

Brain tumors are a major health concern across the globe. They bring high mortality and morbidity rates, as well as harmful symptoms and a reduced quality of life. According to the American Brain Tumor Association, brain tumors account for 15% of all diagnosed cancers, and the number of malignant brains tumors has been increasing in recent years as well. Accurate -diagnosis or brains tumors is essential for effective treatment planning and improving patient outcomes.

Magnetic resonance images (MRI) are a commonly used imaging modalities for the detection and diagnosis of brain tumors. MRIs provide high quality images that give radiologists the ability to precisely identify the tumor's location and characteristics. However, as is the case with all imaging systems, MRI's can be affected by different types of noise and artifacts which reduce the quality of the image. In turn, this makes it more difficult to accurately diagnosis tumors.

With the advances in computational power and in machine learning, there is a lot of potential to utilize machine learning, specifically neural networks, to increase the accuracy and "time to diagnose" of brain tumors. These models can intake large volumes of images and learn the common locations and characteristics of brain tumors. After the model learns what the brain tumors look like, it can then begin to predict and, essentially, diagnose brain tumors with little human intervention. However, the performance of such models can be largely affected by noise and other imaging artifacts, which is not uncommon in clinical practice.

[*]Use footnote for providing further information about author (webpage, alternative address)—*not* for acknowledging funding agencies.

In this project, we aim to investigate the effect of different noises, motion blur, and changes in resolution on the accuracy of a machine learning model trained to classify brain tumors from MRI images. Specifically, we use a data set containing images from three types of tumors, glioma, meningioma, and pituitary, and images with no tumor. Then, we artificially add in noise, motion blur, and resolution changes. The three types of noise we consider are gaussian, Poisson, and quantization.

We use an existing architecture for a convolutional neural network (CNN) that has shown to have high performance as a baseline. Then we add in the synthetic conditions and evaluate performance. Finally, we retrain the CNN and reevaluate performance.

The resulting work of this study can have implications in increasing the accuracy and robustness of machine learning models for brain tumor diagnosis and classification using MRI images. By teaching the machine learning model the locations and characteristics of each type of tumor under harsh conditions, we can improve the accuracy of classification under kinder conditions. Ultimately, we can increase the occurrence of early detections, improve the accuracy of diagnosis, and bring more effective treatment of brain tumors. Hopefully, this should improve patient outcomes and quality of life.

2 Related Works

Brain MRI data has been used in lots of studies to understand how machine learning models can be used to diagnose and classify brain tumors. To start, Soomoro (2023) performed a literature review of how well different machine learning algorithms can detect and identify brain tumors from MRI images. The authors found that deep learning algorithms, specifically CNNs, are more accurate than other algorithms. This is probably due to the flexibility of CNNs and their ability to intake large amounts of image data.

Furthermore, Kidoh et al. (2020) performed another study that evaluated how noise and resolution affect how well a machine learning model can classify brain tumors from MRI images. The authors found that adding gaussian noise affected how accurate the model was at classifying brain tumors. To resolve this, they suggest implementing methods like elastic deformation and rotation to improve the model's ability to handle noise.

In a similar study, Sun et al. (2019) looked at the extent to which noise, motion artifacts, and cropping affected the performance of a CNN when tasked with classifying brain tumors from MRI images. They found that the accuracy was mostly maintained, despite the noise, artifacts, and cropping. This gives us more evidence that a CNN is a good model to use, as well as an architecture to investigate as a possible baseline.

Finally, Nawaz et al. (2021) performed a study that did not investigate adding any noise or artifacts to brain MRI images. Instead, their goal was to optimize classification performance as well as computational efficiency. In other words, they wanted to build a classifier that performed well and that ran quickly and used minimal computing power. They found that the Inception-v3 architecture held up to those standards when compared to other architectures.

In sum, previous studies have paved the way for this study. They have shown that deep neural networks, specifically CNNs, have the power and potential to diagnosis and classify brain tumors very well. However, with the addition of noise and other artifacts, some studies continue to show good results and some studies see their models suffer. The goal of this analysis is to concoct a model that can individually handle each type of noise and artifact as aforementioned.

3 Methods

3.1 Data Preparation

The dataset used in this project is the Brain MRI Images for Brain Tumor Detection, which can be downloaded from the Kaggle.

The dataset contains 253 images of size 240x240 pixels, which were divided into four classes: glioma tumor, meningioma tumor, pituitary tumor, and no tumor. The dataset was randomly split into training (80 percent) and testing (20 percent) sets using the $train - test - split()$ function from the $sklearn$

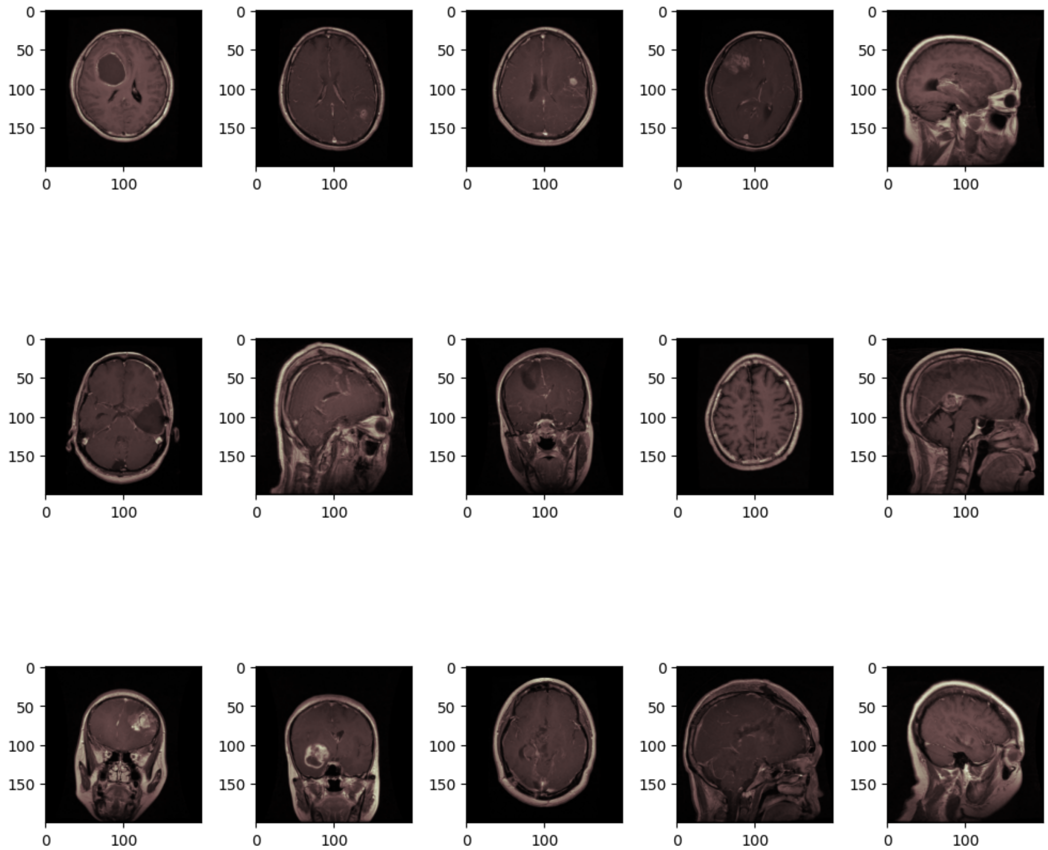


Figure 1: Original Data

Python library. Below are images of the original data and a table summarizing the test/train split and image counts.

Table 1: Quantity of Training and Test Images for Each Tumor Type

Tumor Type	Train Quantity	Test Quantity
Pituitary	1457	300
Glioma	1321	300
Meningioma	1339	306
No Tumor	1595	405

3.2 Baseline Model

We used a ResNet50 model pre-trained on ImageNet as a baseline model. To improve the model's performance, we applied data augmentation to the training set using the $ImageDataGenerator()$ function from the Keras library. The data augmentation techniques used were rotation, width shift, height shift, and horizontal flip. The augmented images were then used to train the model, which was compiled using the Adam optimizer with a learning rate of 0.0001, categorical cross-entropy as the loss function, and accuracy as the metric.

3.3 Gaussian Noise

To investigate the effect of Gaussian noise on the model's performance, we added a small amount of noise to both the training and testing sets using the $np.random.normal()$ function from the

$NumPy$ library. We then loaded the previously trained baseline model and fine-tuned it on the noisy data. The model was compiled using the Adam optimizer, mean squared error as the loss function, and accuracy as the metric. Below is an image of how the Gaussian noise effects the original image.

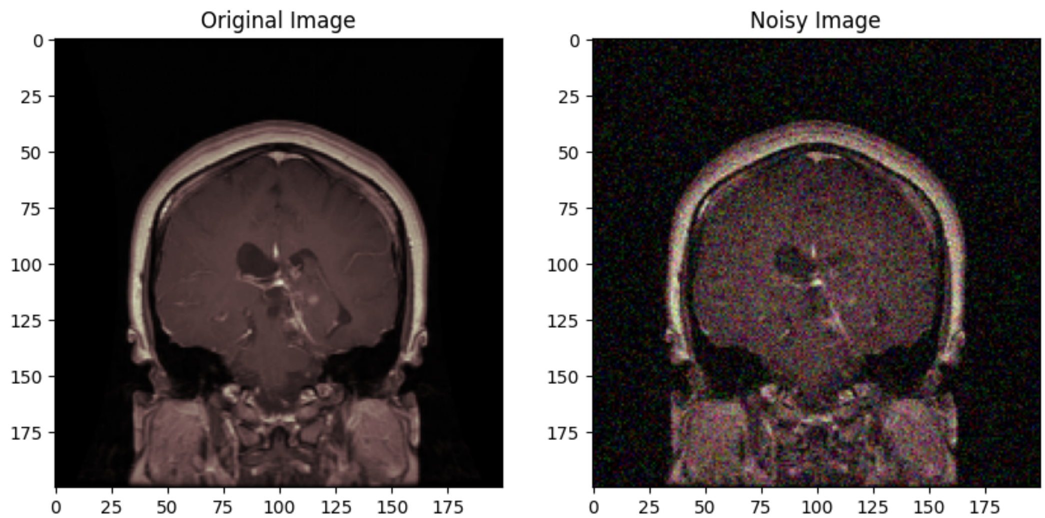


Figure 2: Image with added Gaussian Noise

3.4 Poisson Noise

Similarly, we added a small amount of Poisson noise to both the training and testing sets using the $np.random.poisson()$ function from the $NumPy$ library. We loaded the baseline model and fine-tuned it on the noisy data. The model was compiled using the Adam optimizer, mean squared error as the loss function, and accuracy as the metric. Below is an image with added Poission Noise.

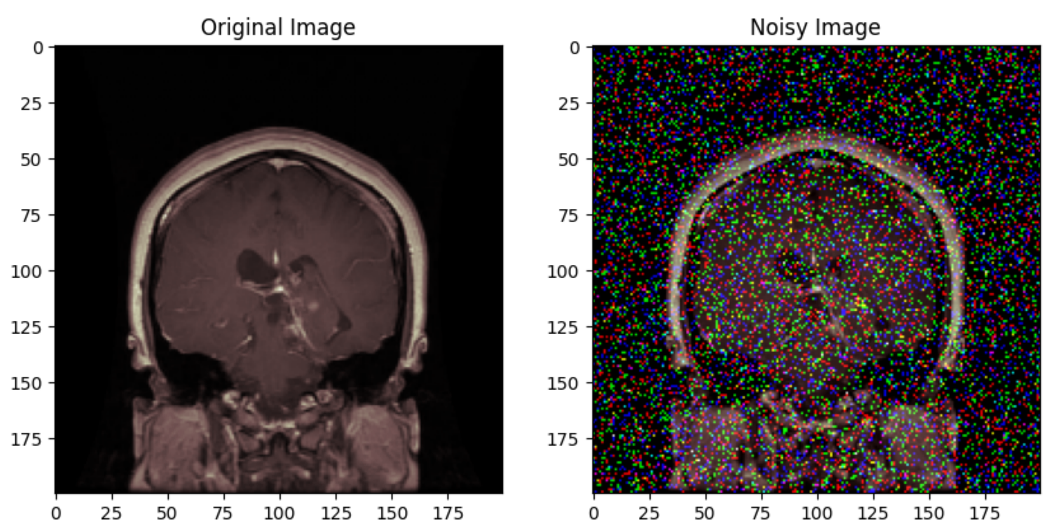


Figure 3: Image with added Poission Noise

3.5 Quantization

To evaluate the effect of quantization noise, we used the $np.floor()$ function from the $NumPy$ library to quantize the pixel values of the images in both the training and testing sets. We then added a small amount of uniform noise to the quantized images using the $np.random.uniform()$ function from the $NumPy$ library. We loaded the baseline model and fine-tuned it on the noisy data. The model was compiled using the Adam optimizer, mean squared error as the loss function, and accuracy as the metric. Below is an example image where quantization noise is added.

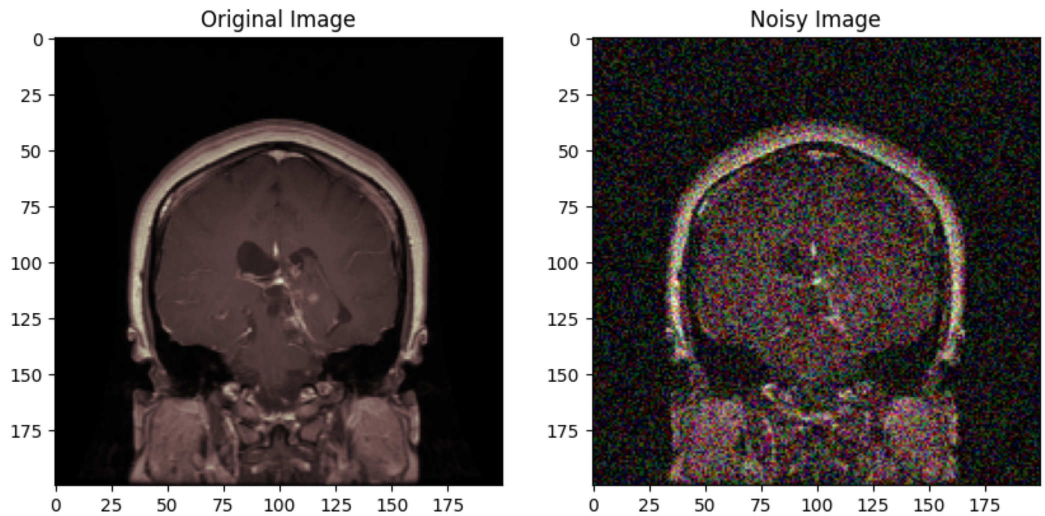


Figure 4: Image with added Quantization Noise

3.6 Motion Blur

Finally, we applied motion blur to the images in both the training and testing sets using a motion blur kernel with a kernel size of 20, applied using the $cv2.filter2D()$ function from the $OpenCV$ library. We loaded the baseline model and fine-tuned it on the blurred data. The model was compiled using the Adam optimizer, mean squared error as the loss function, and accuracy as the metric. Below is an example image where a motion blur is added.

3.7 Resolution

In order to examine the impact of image resolution on the model's performance, we reduced the resolution of both the training and testing sets by a factor of 2 using the $skimage.transform.resize$ function from the $scikit-image$ library. To maintain the quality of the images, we used the "anti-aliasing" option. After downsampling, we added a very small amount of Gaussian noise to the low-resolution training and testing sets using the $np.random.normal()$ function from the $NumPy$ library. This was done to simulate the noise that can be present in real medical images. Next, we loaded the previously trained baseline model and fine-tuned it on the noisy, low-resolution data. The model was compiled using the Adam optimizer, mean squared error as the loss function, and accuracy as the metric. Below is an image that shows what an image looked like after lowering the resolution.

4 Results

In the baseline model, the accuracy achieved on the test set was 98.47%, indicating strong performance in classifying brain tumor MRI images. Below is the Loss Curve, Accuracy Curve, and Confusion Matrix for the baseline model.

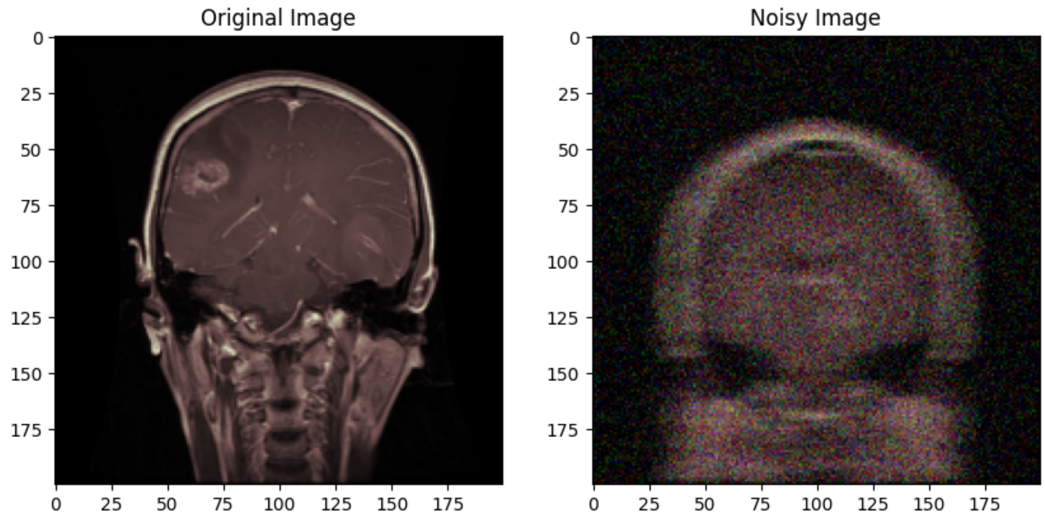
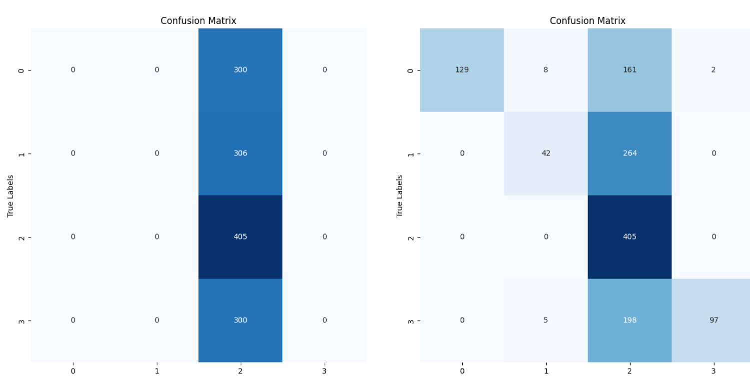


Figure 5: Image with added Motion Blur

In order to investigate the effect of different types of noise on the model's performance, we introduced various types of noise to the test images, including Gaussian noise, poison noise, quantization noise, and motion blur.

For images with Gaussian noise added, the accuracy of the baseline model decreased significantly to 31.5%. However, after fine-tuning on the noisy data, the accuracy was greatly improved, with a final accuracy of 93.83%. Below are confusion matrices showing the performance before retraining and after.

Similarly, when poison noise was added to the test images, the accuracy of the baseline model dropped to 30.89%. After fine-tuning on the noisy data, the accuracy was significantly improved to 94.87%. Below are confusion matrices showing the performance before retraining and after.



(a) Before Retraining (b) After Retraining

Figure 9: Evaluation Graphs for Poisson Noise

When quantization noise was added to the test images, the baseline model achieved an accuracy of 30.89%. However, after fine-tuning on the noisy data, the accuracy was only partially improved to 76.15%. Below are confusion matrices showing the performance before retraining and after.

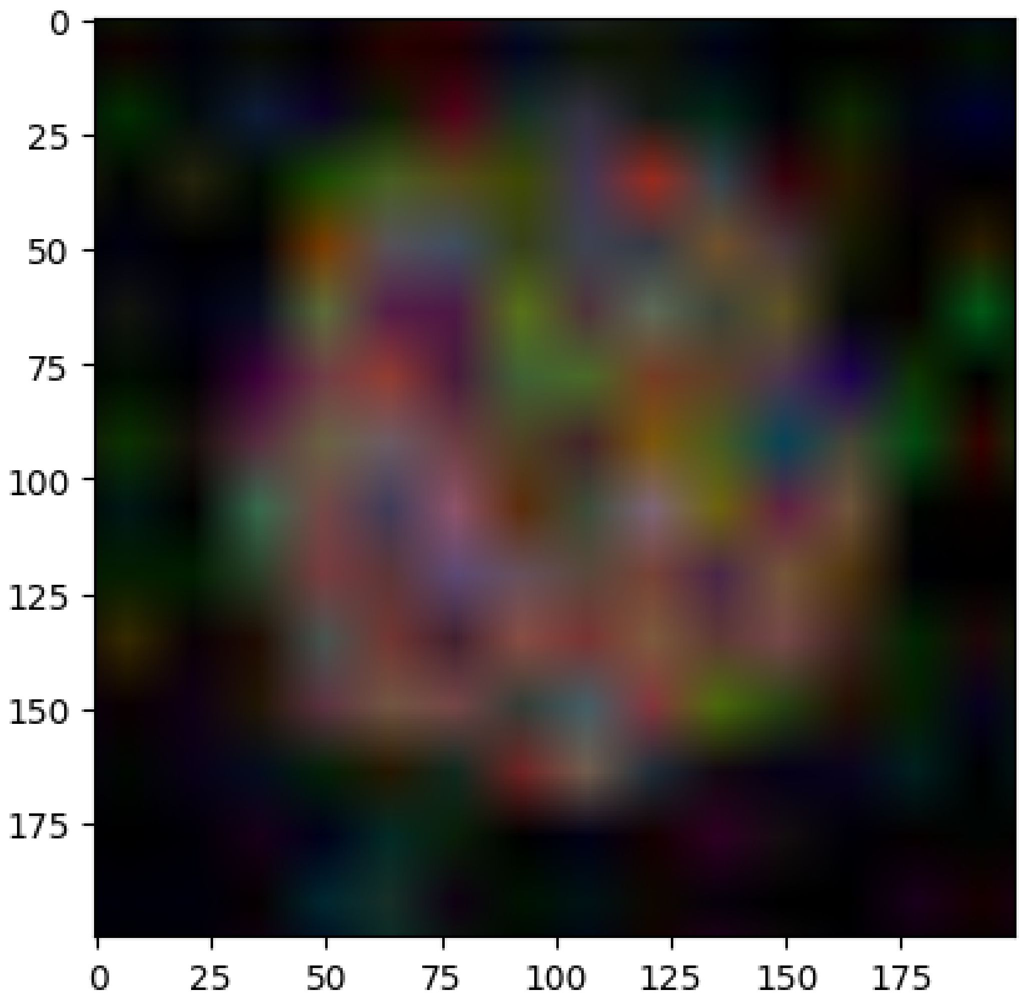
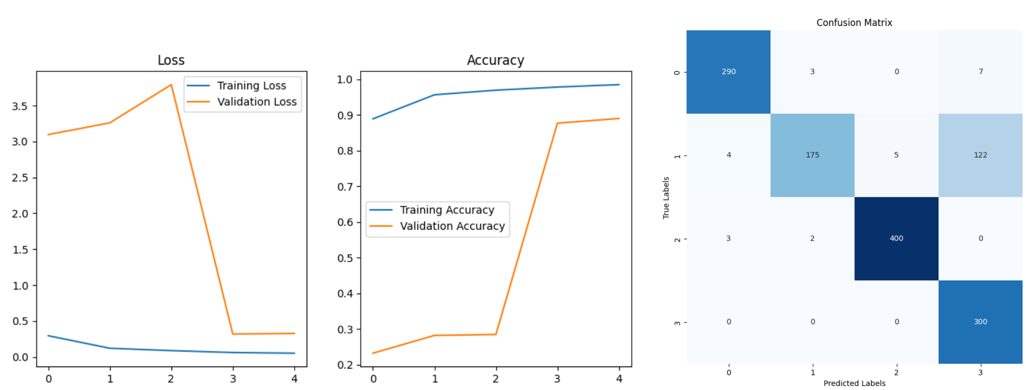
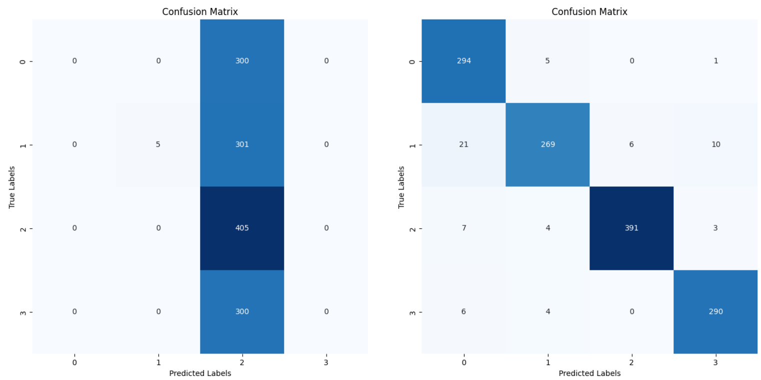


Figure 6: Image with a much lower Resolution.



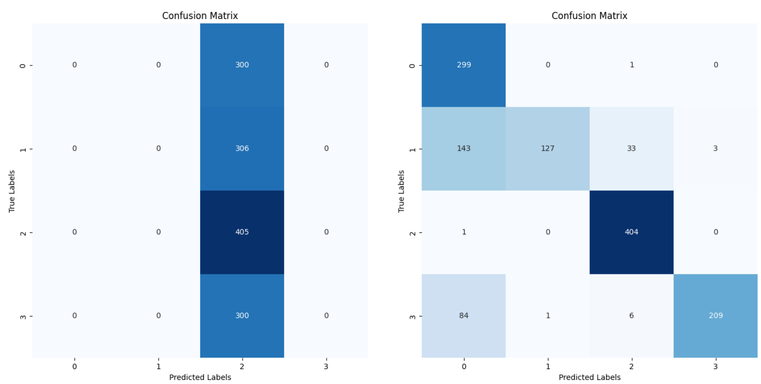
(a) Loss and Accuracy Curves (b) Confusion Matrix

Figure 7: Evaluation Graphs for Baseline Model



(a) Before Retraining (b) After Retraining

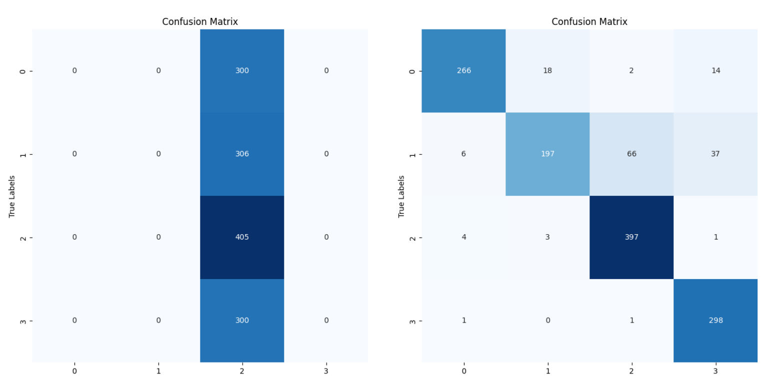
Figure 8: Evaluation Graphs for Gaussian Noise



(a) Before Retraining (b) After Retraining

Figure 10: Evaluation Graphs for Quantization Noise

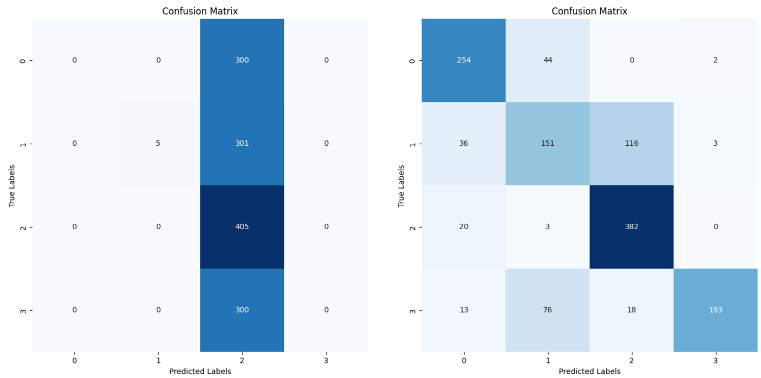
For images with motion blur, the baseline model achieved an accuracy of 30.89%. After fine-tuning on the blurry data, the accuracy was improved to 89.58%. Below are confusion matrices showing the performance before retraining and after.



(a) Before Retraining (b) After Retraining

Figure 11: Evaluation Graphs for Motion Blur

Finally, when the resolution of the images was lowered by a factor of 2, the baseline model achieved an accuracy of 31.27%. After fine-tuning on the lower resolution data, the accuracy was improved to 77.68%. Below are confusion matrices showing the performance before retraining and after.



(a) Before Retraining (b) After Retraining

Figure 12: Evaluation Graphs for Lowered Resolution

These results indicate that fine-tuning on noisy data can significantly improve the performance of a model on noisy images. However, the degree of improvement depends on the type of noise present in the images. In particular, the model was most effective at improving accuracy when the noise added to the images was Gaussian or poison noise, while quantization noise had a more limited impact on performance. Interestingly, before retraining for all added noise or artifacts, the models consistently predicted "no tumor."

5 Conclusion

In this study, we explored the performance of a baseline model for classifying brain tumor MRI images and its ability to adapt to different types of noise. Our results demonstrate that the baseline model achieved strong accuracy on the test set, with an accuracy of 98.47%. However, when noisy data was introduced to the test images, the accuracy dropped significantly.

Fine-tuning the model on the noisy data greatly improved its performance, with accuracy ranging from 76.15% to 94.87%, depending on the type of noise added to the images. Our findings suggest that fine-tuning on noisy data is an effective technique for improving the performance of a model on noisy images.

Furthermore, our study indicates that the type of noise present in the images plays an important role in determining the effectiveness of fine-tuning. Specifically, Gaussian and poison noise had the greatest impact on the model's performance, while quantization noise had a more limited effect.

Overall, our results highlight the importance of considering the impact of noise on the performance of machine learning models and the potential of fine-tuning techniques for improving model robustness. Future studies can explore the effectiveness of fine-tuning on more complex models and additional types of noise.

6 Future Work

One limitation of this study is that we only evaluated the performance of the model on a single dataset of brain tumor MRI images. Future work could involve testing the model's performance on other datasets and image types to see if the findings are generalizable. Additionally, it may be worthwhile to explore the use of other noise reduction techniques, such as denoising autoencoders or generative adversarial networks, to see if they can further improve the model's performance.

Another area of future work could be to explore the impact of different types of noise on the interpretability of the model. While the model was able to adapt to the noisy data, it is unclear how the noise impacted the model's ability to identify important features in the images. Understanding how noise affects the interpretability of deep learning models could be an important area of research in the field of medical image analysis.

Finally, it is important to consider the potential ethical implications of using noisy data to train deep learning models for medical image analysis. While noisy data may improve the model's performance, it could also introduce bias and result in incorrect diagnoses. Therefore, it is important to carefully consider the use of noisy data and to ensure that any potential risks are properly mitigated.

References

[1] T. A. Soomro, L. Zheng, A. J. Afifi, A. Ali, S. Soomro, M. Yin, and J. Gao, "Image segmentation for mr brain tumor detection using machine learning: A review," *IEEE Reviews in Biomedical Engineering*, vol. 16, pp. 70–90, 2023.

[2] M. Kidoh, K. Shinoda, M. Kitajima, K. Isogawa, M. Nambu, H. Uetani, K. Morita, T. Nakaura, M. Tateishi, Y. Yamashita, and Y. Yamashita, "Deep learning based noise reduction for brain mr imaging: Tests on phantoms and healthy volunteers," *Magnetic Resonance in Medical Sciences*, vol. 19, no. 3, pp. 195–206, 2020.

[3] L. Sun, S. Zhang, H. Chen, and L. Luo, "Brain tumor segmentation and survival prediction using multimodal mri scans with deep learning," *Frontiers in Neuroscience*, vol. 13, 2019.

[4] M. Nawaz, T. Nazir, M. Masood, A. Mehmood, R. Mahum, M. A. Khan, S. Kadry, and O. Thinnukool, "Analysis of brain mri images using improved cornernet approach," *Diagnostics*, vol. 11, no. 10, p. 1856, 2021.

[5] M. Nickparvar, "Brain tumor mri dataset," 2021.

[6] American Brain Tumor Association, "About brain tumors." `https://www.abta.org/about-brain-tumors/brain-tumor-education/`, accessed 2023-05-05.

[1] [2] [3] [4] [5] [6]