Tecnológico de Monterrey

Campus Querétaro

Deep Learning - Image Classification

Gamaliel Marines Olvera
A01708746

Advanced AI

October 26, 2024

Introduction

My name is Gamaliel Marines, and in this paper, I explore deep learning solutions for image classification. For this study, I choose to work with a transfer learning of convolutional neural networks with different topologies and hyperparameters.

Objectives and Context of the Problem

Brain tumors are a critical medical issue that can be life-threatening if not diagnosed and treated early. The interpretation of MRI images for tumor detection and classification can be time-intensive and error-prone. A deep learning-based solution offers the potential to support radiologists by automatically classifying brain tumors, which could increase diagnostic speed and accuracy in clinical settings.

Background on the Domain and Problem Being Addressed

The problem lies in reliably differentiating between various types of brain tumors, such as glioma, meningioma, and pituitary tumors, and distinguishing them from normal brain tissues. The dataset used in this study comprises MRI images categorized into different brain tumor classes, including glioma, meningioma, pituitary tumor, and normal cases. This dataset allows the model to learn features unique to each tumor type, supporting accurate classification.

Significance of the Problem in a Real-World Context

A successful model for automated brain tumor classification could improve workflows in radiology, particularly in high-demand environments such as hospitals or specialized oncology centers. Such a model could assist radiologists by highlighting areas of interest and providing initial diagnostic suggestions, enhancing efficiency in diagnosis and potentially improving patient outcomes through faster treatment initiation.

Ethics of the Problem

While automated brain tumor detection offers significant benefits, ethical considerations must be taken into account. Risks of misdiagnosis persist if the model is overly relied upon without proper verification by medical professionals. Additionally, limitations in data diversity or model biases could inadvertently affect the accuracy of tumor classification across different demographics, potentially impacting care equity. Anonymizing data and maintaining patient privacy are crucial in medical applications, and all research involving patient data must adhere to strict ethical guidelines.

Research Questions

Clear definition of the objectives of the study

The primary objective of this project is to develop accurate predictions through the training of convolutional neural networks (CNNs) and to explain their functionality and application in brain tumor classification. Additionally, this project aims to highlight the relevance of these models in the medical field.

Key questions the paper aims to answer

- How effectively can CNNs classify different types of brain tumors from MRI images?
- What features are essential for distinguishing between various tumor types?
- What impact can automated tumor classification have on clinical workflows?

Dataset

Proper data preparation is crucial to developing a reliable deep learning model, as it ensures the model is trained on high-quality data, directly impacting performance and accuracy.

Data Collection and Description

It is very important to choose and have an adequate dataset which allows to have a great performing model. I faced the obstacle of not having a good dataset and that delayed my work. I had chosen a dataset to detect bone fractures; however, the dataset did not contain a division of the areas of the body which had suffered a fracture, therefore making it very difficult to have a multiple classifier, forced me to have a binary classifier and settle down for classifying fractured and not fractured. Since the dataset contained x-rays for hands, armas, legs, hips, collar bones, etc and the dataset had more than nine thousand images it was impossible for me to properly clean the dataset and reach the accuracy I was looking for. This made me realize the importance of choosing the right dataset and to review its characteristics.

Sources of the Dataset and Its Main Features

The data used in this study originates from a brain tumor classification MRI dataset available on Kaggle, shared by Sartaj. The dataset, "Uncovering Knowledge: A Clean Brain Tumor Dataset for Advanced Medical Research," includes MRI images classified into four categories: glioma tumors, meningioma tumors, pituitary tumors, and normal brain tissues.

Dataset Structure

The dataset is organized into folders by category (glioma, meningioma, pituitary tumor, and normal) with a total of 3,096 images across the four classes. The dataset is licensed under CC0, making it accessible to the medical research community to promote collaboration and innovation.

Methods for Handling Image Quality, Augmentation, and Scaling

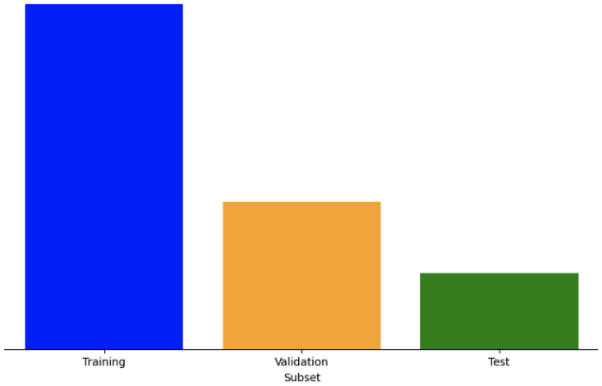
Data cleaning includes preprocessing images to standardize their format, enhance data diversity through augmentation, and rescale pixel values to ensure consistency across samples:

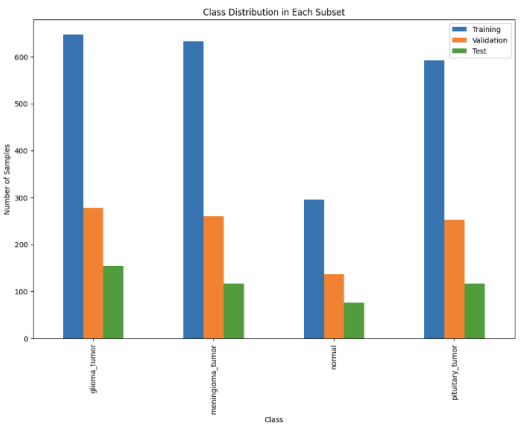
- **Rescaling**: Pixel values are normalized by dividing by 255 to transform them into the [0, 1] range, which aids in faster and more stable model convergence.
- **Data Augmentation**: To increase data diversity and improve generalization, images in the training set undergo transformations such as random rotations (up to 20 degrees), horizontal flips, width and height shifts, shearing, and zooming.

Dataset Partitioning

Once preprocessed, the data is partitioned into training, validation, and test sets:

- **Training Set**: 2168 images for model training and learning.
- **Validation Set**: 928 images for hyperparameter tuning and performance monitoring, helping to avoid overfitting.
- **Test Set**: 464 images for final evaluation on unseen data, representing the overall distribution of fractured and non-fractured cases.



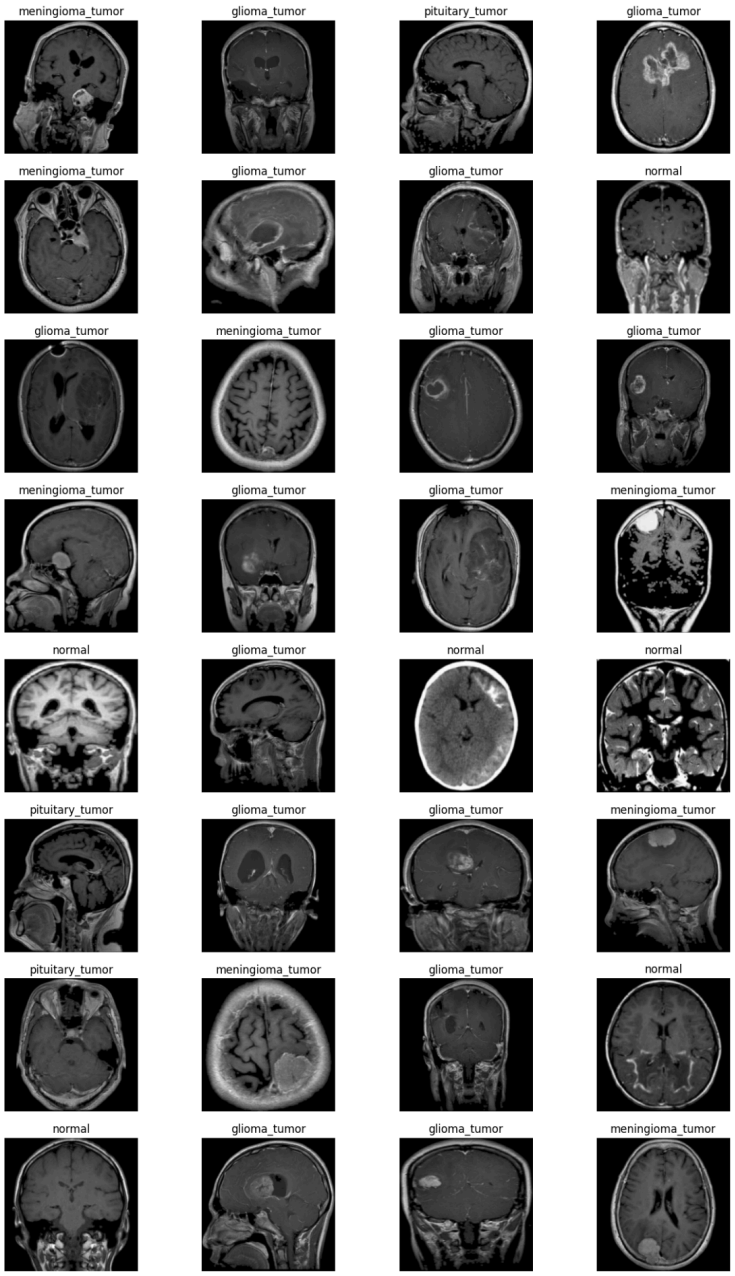


Explanation of Decisions on Image Processing

Data points (images) are augmented only in the training set to improve model robustness, while validation and test sets remain unaugmented to provide accurate performance evaluation. This decision ensures that model performance on unaltered data can generalize well to real-world scenarios.

Dataset Quality Review

To verify the quality of the dataset and the labeling, I show a portion of the dataset which makes it easier for external reviewers to understand the problem and professionals of the area to review the correctness of the data set.



Models

Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) are excellent for image classification and other visual tasks. They automatically learn and detect patterns in images, like edges, textures, and complex shapes.

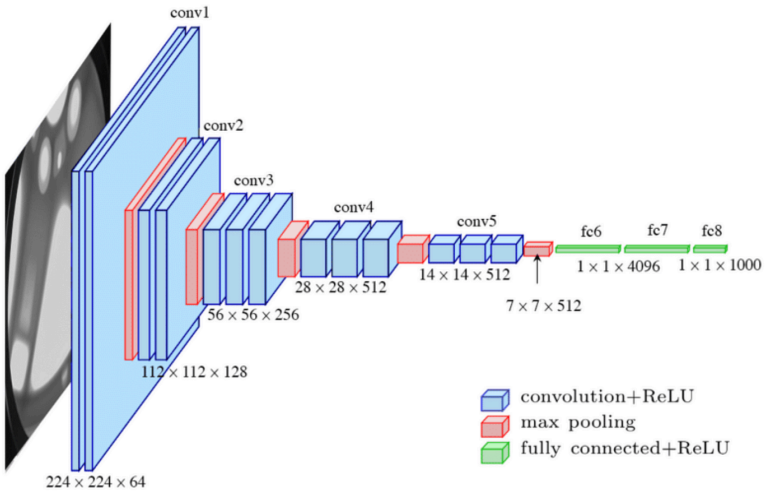
Main Layers in CNNs

- **Convolutional Layers**: These layers apply filters (small grids of weights) to the image, scanning for local features like edges and color gradients. The filters extract important features by focusing on small regions of the image, building progressively more complex representations as they move through the network.
- **Pooling Layers**: Pooling layers (often max pooling) reduce the spatial dimensions of the data. This reduces the computational load, retains dominant features, and introduces translation invariance, helping the network recognize patterns even if they shift within the image.
- **Fully Connected Layers**: These layers aggregate the learned features to produce the final output. They connect every neuron from the previous layer, facilitating classification by combining the extracted features.

VGG16

VGG16 is a classic CNN architecture known for its simplicity and depth, which made it a breakthrough in deep learning for image classification. It consists of 16 layers with weights (hence the "16") and is characterized by:

- **Multiple Convolutional Layers**: Each convolutional block has two or three convolutional layers with small 3x3 filters, followed by a max-pooling layer to downsample.
- **Uniform Design**: VGG16 consistently uses 3x3 filters and 2x2 pooling windows, simplifying the model architecture and making it easier to train.
- **Performance**: VGG16 achieves high accuracy on large image classification datasets (like ImageNet), but it is computationally demanding due to the large number of parameters, making it suitable for high-performance hardware.



(Le, 2022)

ResNet

ResNet, short for Residual Network, introduced a groundbreaking concept to tackle the limitations of very deep networks. It uses "skip connections" or "residual connections" that allow gradients to bypass certain layers, making it easier to train very deep networks without encountering vanishing gradient issues.

- **Residual Blocks**: ResNet's defining feature is the residual block, which adds a shortcut connection, allowing the network to skip over layers. This solves the issue of vanishing gradients and allows for models with hundreds of layers.
- **Depth without Degradation**: ResNet architectures can be much deeper (e.g., ResNet-50, ResNet-101) than traditional CNNs, capturing more complex features while avoiding the degradation problem where adding more layers can worsen performance.
- **Efficiency and Accuracy**: ResNet architectures outperform previous deep networks on benchmarks, balancing depth with efficient training, and are widely used in complex image classification and detection tasks.

Implementation

Code Structure Overview

The project includes data preprocessing, model building, training, evaluation, and predictions.

1. **Data Loading and Preprocessing**
    - Load dataset
    - Scale the data

    - Define batch size: 32
    - image size: (224, 224)
2. **Dataset Partitioning**
    - Split into training (70%), validation (15%), and test sets (15%) for robust training.
3. **Model Building**
    - **VGG16:** Built using transfer learning.
    - **ResNet:** Built using transfer learning.
4. **Model Training and Evaluation**
    - Train and evaluate both models, generating confusion matrices, accuracy and loss plots, and classification reports.
5. **Predictions**
    - Predictions with the test subset.
    - Predictions with external images.
6. **Results Comparison**
    - Compare metrics (accuracy, precision, recall, F1-score) of both models.

Libraries Used

- **Data Handling and File Management**:
    - `os` and `shutil` for directory and file management, aiding in organizing and manipulating image datasets.
    - `pandas` (`pd`) and `numpy` (`np`) for data handling and manipulation, allowing efficient processing of dataframes and numerical arrays.
- **Data Visualization**:
    - `matplotlib.pyplot` (`plt`) and `seaborn` (`sns`) for creating visual representations of data distributions, model performance, and confusion matrices, aiding in comprehensive analysis.
- **Modeling and Evaluation**:
    - `scipy` and `scikit-learn` for utilities such as data splitting (`train_test_split`) and evaluation metrics like `classification_report` and `confusion_matrix`.
    - `tensorflow` and `keras` for building and training deep learning models, using layers such as `Conv2D`, `MaxPooling2D`, and fully connected layers (`Dense`) for image classification.
    - `ImageDataGenerator` for data augmentation, and callbacks such as `ReduceLROnPlateau`, `EarlyStopping`, and `ModelCheckpoint` to optimize training.
    - `mlxtend.plotting` for additional visualization support with `plot_confusion_matrix`, enhancing performance analysis.
- **Image Processing**:

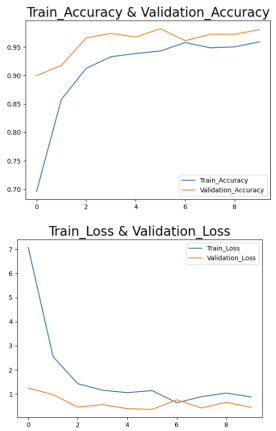
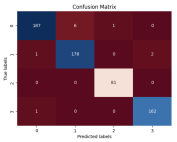
  - `PIL` (`Image`) for loading and processing image files, ensuring they are ready for model training.
- **Warnings Management**:
  - `warnings` to filter out unnecessary warnings during training, keeping the output focused on critical information.

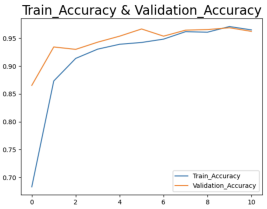
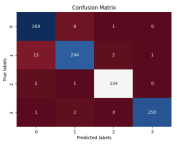
Evaluation Metrics

- **Accuracy:** Overall correctness.
- **Precision:** Ratio of true positives to all predicted positives, crucial if false positives are costly.
- **Recall:** Ratio of true positives to actual positives, key when false negatives are costly.
- **F1-Score:** Harmonic mean of precision and recall for balanced evaluation.
- **Confusion Matrix:** Visualizes true positives, true negatives, false positives, and false negatives.
- **Loss Plot:** Tracks training and validation loss over epochs to detect overfitting.

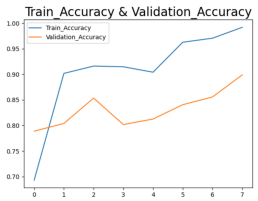
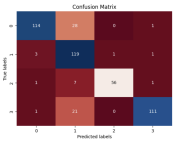
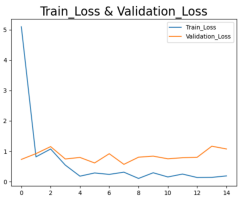
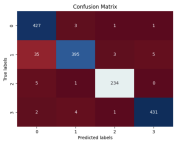
Results and Discussion

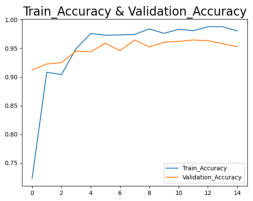
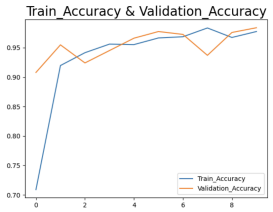
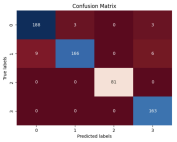
VGG16 Result Table

Sampling	Hyperparameters and parameters	Accuracy	Acc & Loss Plot	Confussion Matrix (CM)	Interpretation
3096 Images 2168 training 928 validation 464 test (No data augmentation)	dropout: 0.5 activation: softmax for 4 classes check point: save the best model early stopping: patience 5 optimizer: adam loss: sparse categorical cross entropy epochs: 10	training: 0.96 val: 0.98 test: 0.98			The drop out is important to prevent overfitting and the model over adjusting and memorizing instead of learning, the activation for 4 classes is important because I am predicting 4 different classes, the checkpoint is to save the best performing version of the model. The accuracy of the model is great. The accuracy of test

					and validation is better than training because of the size of the subset and therefore can face less extreme datum. The acc and loss plots show that there is no over nor underfitting and the confusion matrix confirms that the model is well trained and can make accurate predictions. The next step is to test how can the accuracy improve when there is data augmentation and more number of epochs.
Training set batches: 68 Validation set batches: 19 Test set batches: 10 data augmentation: rotation: 0.1 zoom: 01	dropout: 0.5 activation: softmax for 4 classes check point: save the best model early stopping: patience 5 optimizer: adam loss: sparse categorical cross entropy epochs: 12	training: 0.97 val: 0.97 test: 0.97	 		with data augmentation the accuracy of test, validation and test are the same as decimals. This means that with more epochs I could reach even a better performance.

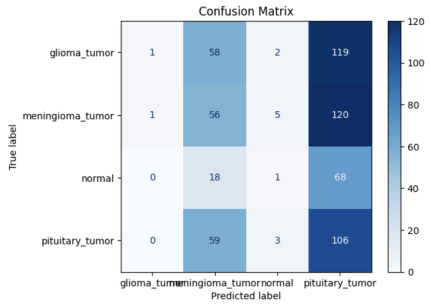
ResNet Result Table

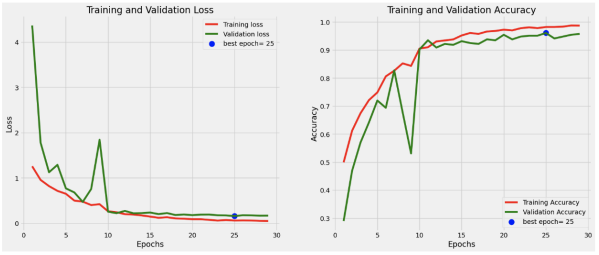
Sampling	Hyperparameters	Accuracy	Acc & Loss Plot	Confussion Matrix (CM)	Interpretation
3096 total Images 2168 training images 928 validation images 464 test images	activation: softmax for 4 classes check point: save the best model optimizer: adam loss: sparse categorical cross entropy epochs: 8	training acc: 95% validation: 89% test accuracy: 88%	 		without dropout and without early stopping the model reaches a very good accuracy for the training set. However, the validation accuracy is way less than the training subset. This means that the model is overfitting and memorizing for the training subset. With the confusion matrix it is obvious that the model is over adjusting for certain classes and therefore there is confusion with the predictions of class 1. To solve these issues (overfitting and confusion with the predictions of classes) I will “tune the settings” of the model.
3096 total Images 2168 training images 928 validation images 464 test images	dropout: 0.8 activation: softmax for 4 classes check point: save the best model optimizer: adam loss: sparse	training acc: 97% validation: 96% test accuracy: 96%			By adding dropout I solve the problem of overfitting and can train the model longer (for more epochs) and increase the accuracy of the predictions. The accuracy

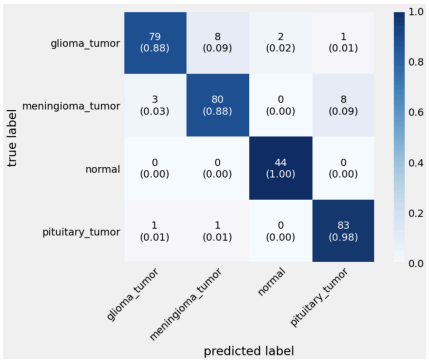
	categorical cross entropy epochs: 15				levels of the subsets and the confusion matrix confirms that the model indeed improved.
3096 total Images 2168 training images 928 validation images 464 test images	dropout: 0.3 activation: softmax for 4 classes check point: save the best model early stopping: patience 5 optimizer: adam loss: sparse categorical cross entropy epochs: 10	val_accuracy: 0.9838 Test Accuracy = 0.9660742 accuracy: 0.9792	 		To reduce the number of epochs needed I reduced the dropout rate and added the early stopping with patience of 5 epochs. With these changes I reach a similar accuracy but with less epochs needed. The accuracy levels, the difference of the training accuracy and validation/test accuracy confirms that the model is not overfitting and the confusion matrix reaffirms the correctness of the predictions made by the model.

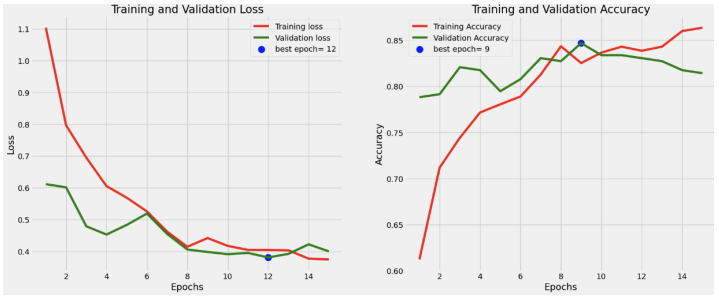
Benchmarks and Where do I stand

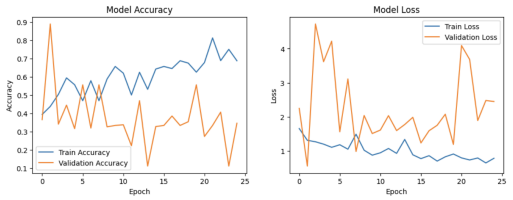
As previously mentioned, the dataset I am using originates from Kaggle. Below are some examples of results obtained from various sources and implementations using the same dataset. Out of respect for their work and privacy, I have chosen not to cite them.

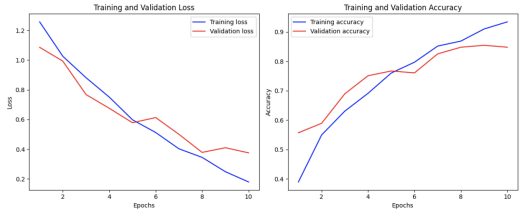








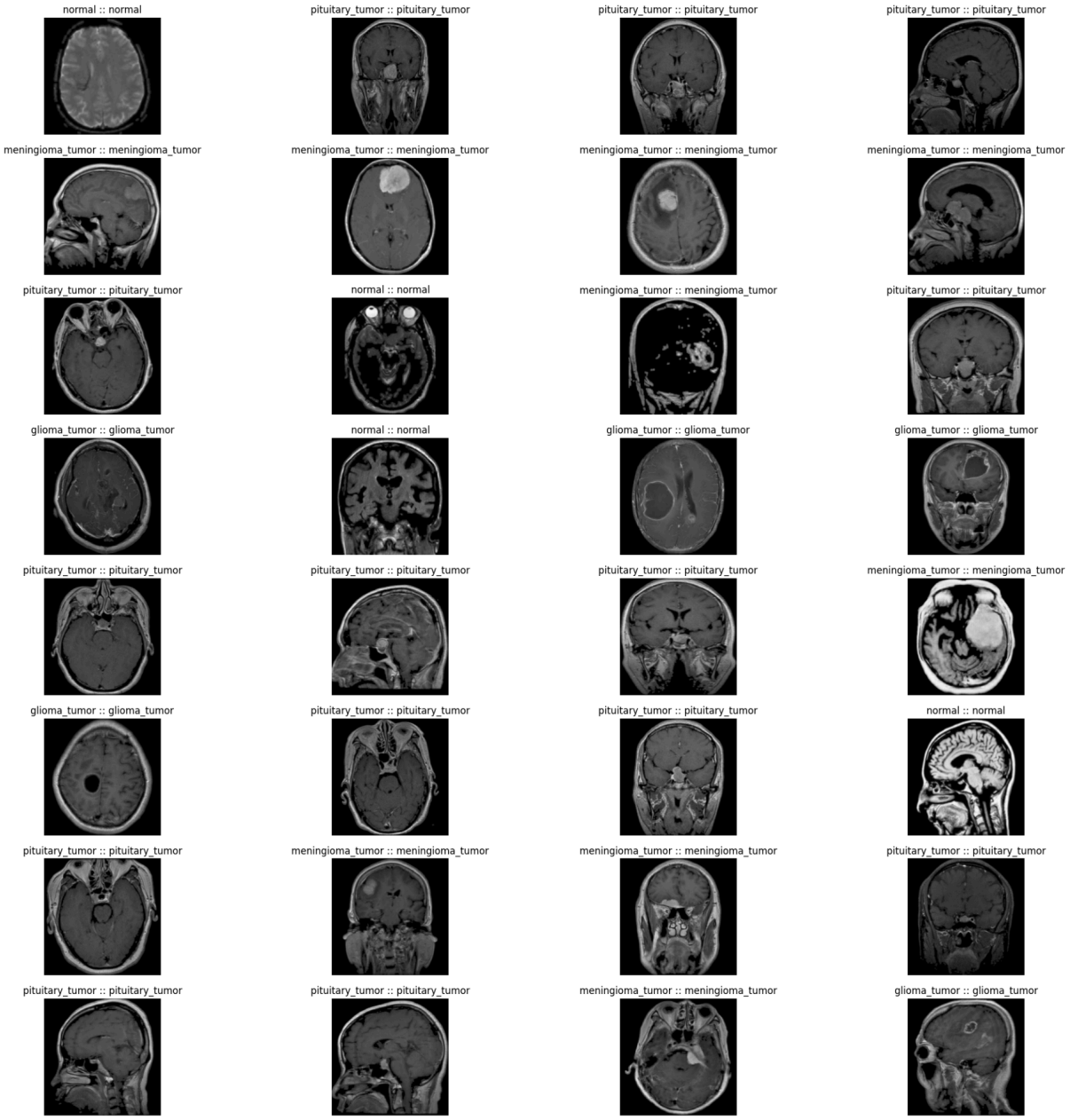




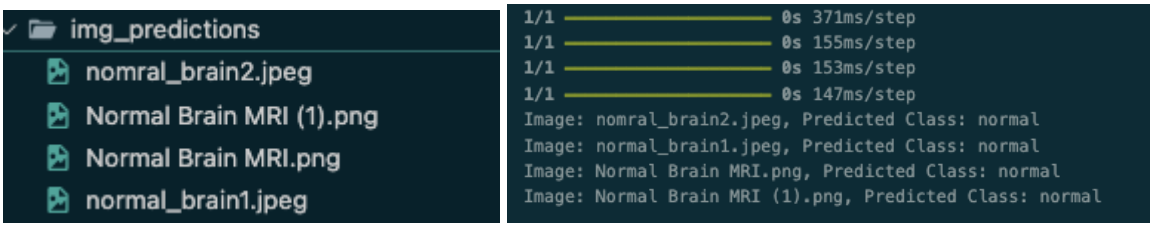
The average accuracy achieved is approximately 92%, with a highest value of 97% and a lowest of 40% on the validation subset (noting that some sources do not include a test subset). Comparing my results to others, it is evident that I have achieved a more than acceptable level of accuracy.

Predictions

I make different types of predictions. The first table of predictions is composed of images from the original dataset, from the subset “test”. This table consists of showing the image, the label of the images and the prediction made. This format allows to verify the accuracy of the model and its predictions.



The second prediction is made with images external of the dataset, since I have the control and now their label I just print the prediction made by the model:



Interpretation of Results

All of the approaches are having great performance and have great accuracy, precision, recall, and F1-score. This is because the models have already been trained and are just learning based on the specific features of the brain MRI dataset. VGG16 and ResNet are highly effective because they have been pre-trained on large image datasets, which enables them to recognize complex patterns and features in images.

Insights Gained from the Model's Performance

ResNet (Residual Networks) is generally preferred over VGG16 in many image classification tasks because of its architectural innovations, especially when handling deeper networks. Here are the main reasons:

1. **Residual Connections**: ResNet introduces "skip connections" or "residual connections," which allow the network to skip layers by adding the input directly to the output of a few stacked layers. This helps prevent the "vanishing gradient" problem by making it easier to propagate gradients through the network, allowing ResNet to go much deeper (e.g., 50, 101, or 152 layers) without performance degradation. VGG16, in contrast, is limited to 16 layers and often struggles when deeper layers are added due to gradient vanishing or exploding issues.
2. **Efficiency in Training**: Because of these skip connections, ResNet converges faster and with fewer training errors in deep layers compared to VGG16. This faster convergence allows for better utilization of resources during training and improves overall performance.
3. **Model Depth**: ResNet achieves greater accuracy by enabling the use of more layers without causing degradation, meaning it can capture more complex features. VGG16, being shallower in comparison, is less effective in capturing such deep hierarchical features.
4. **Computational Efficiency**: ResNet models, particularly ResNet-50, typically require fewer parameters than VGG16 (ResNet-50 has about 23 million parameters, while VGG16 has about 138 million). This reduced parameter count makes ResNet more computationally efficient in both training and inference, especially for memory-limited hardware.
5. **Generalization Performance**: ResNet's ability to generalize well across tasks has made it the backbone of choice for transfer learning. Its skip connections allow it to generalize to a broader range of image classification and feature extraction tasks, making it versatile and highly effective for various applications, including medical imaging.

Conclusion

In conclusion, this study highlights the potential of deep learning, particularly convolutional neural networks, to advance brain tumor classification from MRI images, offering valuable support in clinical settings. By comparing VGG16 and ResNet architectures, the findings

demonstrate the effectiveness of transfer learning and the importance of model selection based on computational efficiency and predictive accuracy. While VGG16 performs well, ResNet's skip connections and depth allow it to address complex patterns more effectively, enhancing its robustness in image classification tasks like brain tumor detection. With ResNet showing superior performance, further improvements, such as increased training epochs and optimized data augmentation, could push accuracy even higher. The success of this model underscores the viability of automated MRI interpretation as a tool to aid radiologists, potentially speeding up diagnosis and improving patient care.

References

Le, K. (2022, 7 enero). An overview of VGG16 and NiN models - Khuyen Le - Medium. *Medium*. https://lekhuyen.medium.com/an-overview-of-vgg16-and-nin-models-96e4bf398484

tf.keras.applications.VGG16. (s. f.). TensorFlow. https://www.tensorflow.org/api_docs/python/tf/keras/applications/VGG16

tf.keras.applications.ResNet50. (s. f.). TensorFlow. https://www.tensorflow.org/api_docs/python/tf/keras/applications/ResNet50