ISSN 1870-4069

Detection and Classification of Brain Tumors in MRIs Using CNNs

Brandon Alejandro Llaca-Sánchez, Dulce Sofía Marín-Chávez,
Alejandra Vilchis-Yubi

Universidad Autónoma de Querétaro,
Facultad de Ingeniería, Campus Aeropuerto, Santiago de Querétaro,
Mexico

{bralig14, dulcesofia021098, alevyubi}@gmail.com

Abstract. This paper describes implementing a Convolutional Neural Network (CNN) model for detecting and classifying brain tumors from Magnetic Resonance Images (MRIs) using TensorFlow and Keras. The model classifies images into one of 4 possible classes: glioma, meningioma, pituitary, or "no tumor" (healthy patient). The data augmentation paradigm was used alongside image processing techniques to expand the training dataset, achieving a final test set accuracy of 89%. Brain tumors present challenges in their detection and classification due to their variability in shape, size, and location, which complicates medical diagnosis using traditional methods promptly. To address this challenge, this study employs a CNN model that integrates convolutional layers alternated with pooling layers, inspired by modifications of existing architectures that have proven to be efficient in terms of the computational cost-accuracy ratio. This work aims to refine the accuracy of classification among different types of tumor, but also versus non-tumor images. Furthermore, a user-friendly Python-based graphical interface has been developed to enable users unfamiliar with deep learning models to conduct preliminary MRI classifications, potentially saving diagnostic time and resources in medical environments.

Keywords: Brain tumor, convolutional neural network, MRIs, data augmentation, graphical interface.

1 Introduction

This paper applies a deep learning model (using Convolutional Neuronal Networks, CNNs) with data augmentation and image processing techniques, to detect and classify brain tumors in magnetic resonance images, using Tensorflow and Keras. Brain tumors pose a challenge in their detection and classification due to their variability in shape, size, and location, making timely medical diagnosis difficult using conventional techniques. In response to this problem, a convolutional neural network-based model inspired by modifications of existing architectures was used in this study and was shown to be efficient in terms of computational cost/accuracy ratio to improve the accuracy of classification between tumor types and non-tumor cases.

pp. 5–18; rec. 2024-06-02; acc. 2024-08-12

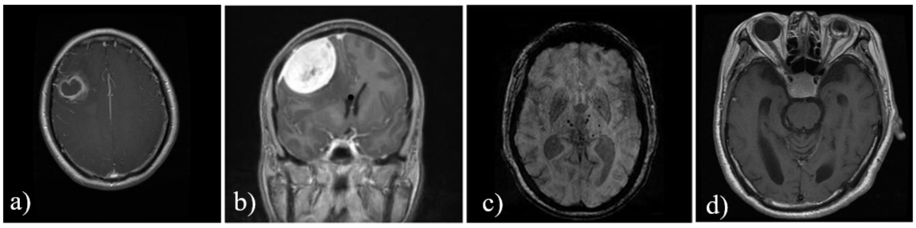


Fig. 1. Examples of MRIs of each class, a) Glioma, b) Meningioma, c) No tumor, d) Pituitary.

To balance the variability of clinical images and avoid the need for more data, data augmentation techniques such as geometric transformations and intensity adjustments are applied. The transformations include partial rotations, inversions, and contrast adjustments, which give the model the ability to identify tumors under different visual conditions and increase its robustness to images from new patients.

TensorFlow was used to create and train CNN models tailored to the specific characteristics of MRIs. The networks were trained with an extended dataset containing images of gliomas, meningiomas, pituitary tumors, and healthy patients, all normalized to a uniform format of 250×250 pixels to standardize the inputs to the model. The present research focuses on improving the accuracy of brain tumor classification using CNNs and data augmentation techniques. The results obtained could then be a useful tool for medical professionals to improve the efficiency of brain tumor diagnosis.

2 Related Work

There are many research papers on the classification of brain tumors. Some of them are Mohsen [10], who used a Deep Neural Network Classifier with a dataset of 66 classes of brain tumors (glioblastoma, sarcoma, and metastatic bronchogenic carcinoma) and normal MRIs using a discrete wavelet transform (DWT) and principal component analysis (PCA) with a classification rate of over 93%. Rai [13] presents a deep neural network called U-NEt (LU-Net), which distinguishes between normal and abnormal MRI images of the brain using a data set of 253 images and achieves an accuracy of 88%. A Convolutional Neural Network with a Long Short Term Memory (LSTM) was used by the authors in [16] for feature extraction to augment the CNN extraction features with a dataset of 3264 MRI scans.

Nyoman Abiwinanda [6] trained different CNN architectures such as AlexNet, ResNet, and VGG16 to detect 3 types of brain tumors (glioma, meningioma, and pituitary). The dataset used consists of 3064 T-1 weighted CE-MRI images. The best-performing architecture had a training accuracy of 98.51% and a validation accuracy of 84.19%. Yakub Bhanothu [4] proposed an algorithm called ”Faster R-CNN”, which is also used to classify and detect the 3 types of brain tumors mentioned above. The algorithm uses VGG-16 as the base layer and consists of three blocks called RPN, a region of interest (RoI) pooling and a regional-based convolutional neural network. The dataset used is public, it contained 805 MRIs for glioma, 694 for meningioma, and 907 for tumors. They achieved an average precision of 77.60%.

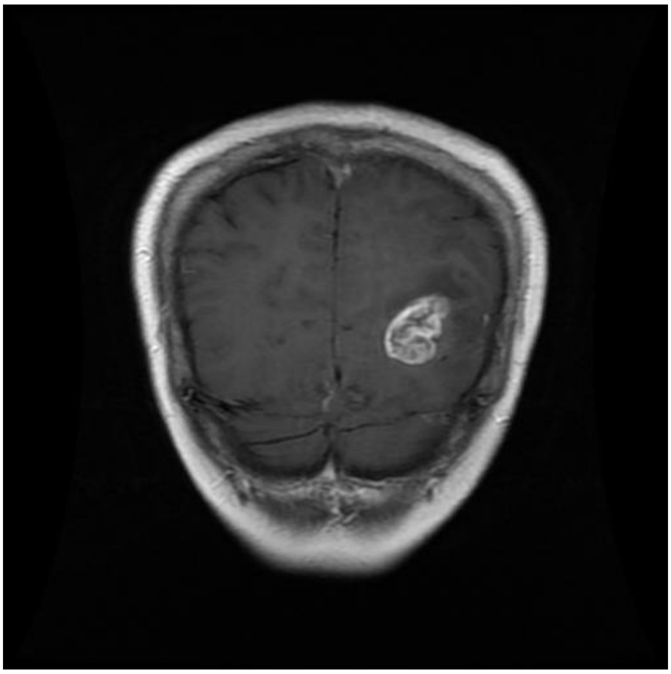


Fig. 2. MRIs of Glioma tumor example.

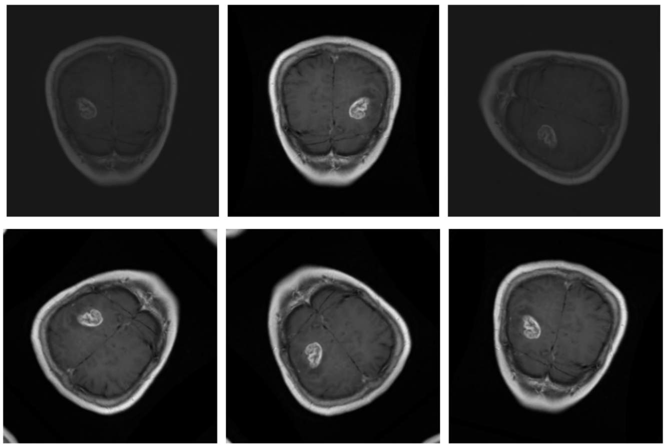
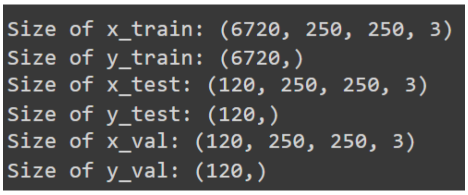


Fig. 3. Example of the 6 transformations applied to the side glioma image.

An interesting approach was taken by the authors in [2], where they first used several filters to preprocess the MRIs, aiming to differentiate the various elements of the image more easily. This preprocessing step facilitated the clustering and segmentation of tumors. A prediction of tumor versus non-tumor was then made using a stacked sparse autoencoder (SSAE) model with two fine-tuned layers, obtaining an accuracy between 90-100% on the BRATS (Brain Tumor Segmentation) datasets [3].



```
Size of x_train: (6720, 250, 250, 3)
Size of y_train: (6720,)
Size of x_test: (120, 250, 250, 3)
Size of y_test: (120,)
Size of x_val: (120, 250, 250, 3)
Size of y_val: (120,)
```

Fig. 4. Sizes and dimensions of training, validation, and test sets.

Regarding data augmentation methods, in [8] Han et al. leveraged Progressive Growing of GANs (PGGANs) to generate synthetic 256x256 MRIs. This method, which was evaluated on the BRATS 2016 dataset [9] and combined with traditional data augmentation methods, improved brain tumor detection accuracy to 91.08%, with 86.60% of sensitivity and 97.60% of specificity, delivering promising results for clinical use. The main goal of this research, which focuses on detecting and classifying brain tumors, is to apply these models in software or interfaces that are user-friendly for the medical staff. Ucuzal [15] developed a free web-based on deep learning that can be used in the detection of brain tumors, specifically Glioma, Meningioma, and Pituitary tumors on MRIs. They used a Keras library to build a deep learning model and achieved 95% accuracy in the test dataset.

However, for further studies, they propose to include the classification and detection of MRIs of healthy patients [15]. An extensive review of more related works in the field (until 2021) is provided in [11]. Al-Zoghby, who proposed the DCTN model using a CNN with VGG-16 architecture and a dataset with 233 patients, achieved 99% accuracy [1] during testing. Özkaraca used DenseNet, VGG-16, and modified CNN architectures, reaching an accuracy of up to 92%.

However, in the confusion matrix for the VGG-16 model, the values obtained showed many mistakes in glioma classification [12]. Srinivasan presented a comparison of AlexNet, DenseNet121, ResNet-101, VGG-19, and GoogleNet models, reaffirming the superiority of CNN in the field of brain tumor classification [14]. As reviewed by many authors, significant results have been achieved in detecting and classifying brain tumors using CNNs. Two of the main contributions of this investigation are to implement and continue exploring the data augmentation approach for this problem, as well as to develop a graphical Python interface that is easy for medical staff to use and understand, in order to assist them in making fast and appropriate diagnoses.

3 Methods

The methodology of this study involved various phases, from data preparation to the training and evaluation of the neural network. The steps taken to implement the model successfully are described below:

3.1 Dataset

The study was performed using the public database ”Brain MRI Scans for Brain Tumor Classification” obtained from the platform ”Kaggle” [7]. The dataset contains

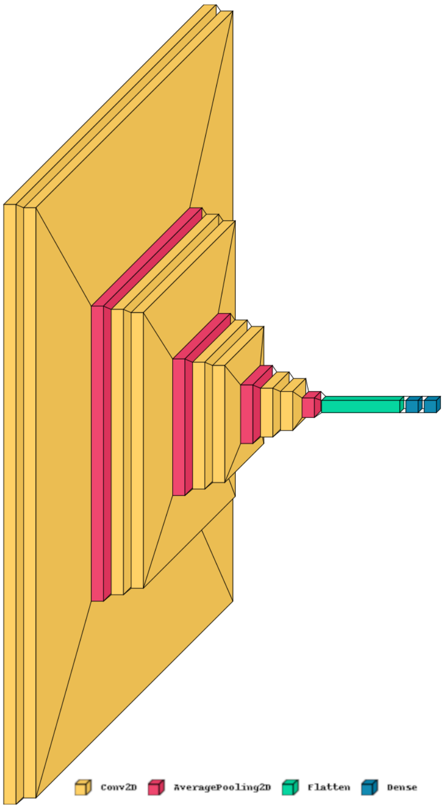


Fig. 5. Three-dimensional representation of the network architecture. Yellow layers: Convolutional layers; Red layers: Pooling layers; Green layer: Flatten layer; Blue layers: Dense layers.

high-quality Brain Magnetic Resonance Imaging scans with diverse tumor types. It is classified into four classes and each image is labeled with one of these classes: “Pituitary”, “Glioma”, “Meningioma” and “No tumor”. It has a total number of images of 1311, where 300 images belong to Pituitary, 306 images to Meningioma, 300 images to Glioma, and 405 images to No tumor class. Fig. 1 shows an example of MRIs that belong to each class.

3.2 Data Processing

To achieve a balanced distribution of classes, it was first decided to select the first 300 available images of each class to homogenize the number of data per class, since Meningioma and No tumor classes have more than 300 images.

All images were uploaded to the working environment, 300 images for each class. The image set for each class was divided into 240 images for training about 30 images for testing and 30 for validation. The selection of images for each group was randomized. The total number of images for the training process was 960 images, and

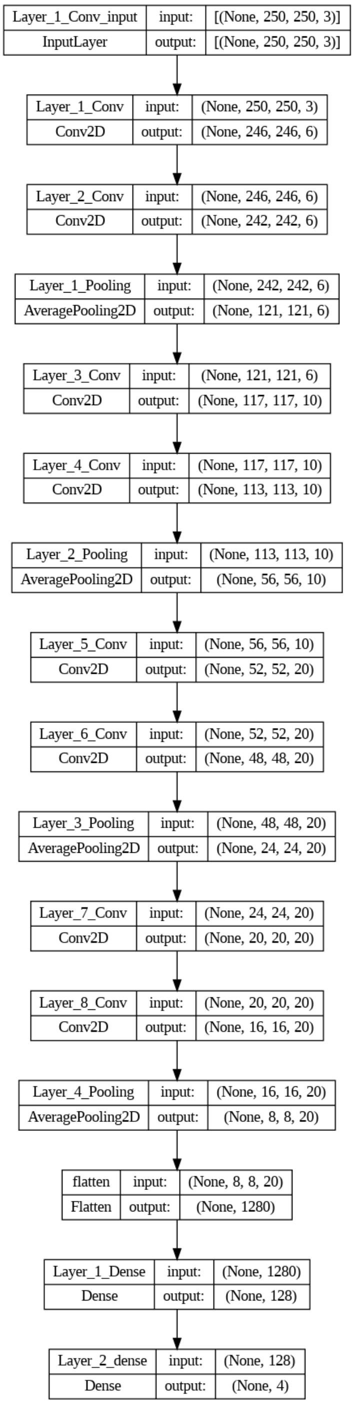
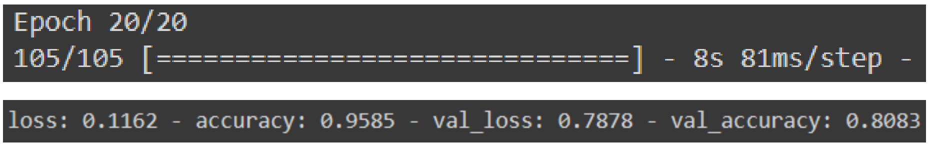

Layer_1_Conv_input | input: | [(None, 250, 250, 3)]
InputLayer | output: | [(None, 250, 250, 3)]
Layer_1_Conv | input: | (None, 250, 250, 3)
Conv2D | output: | (None, 246, 246, 6)
Layer_2_Conv | input: | (None, 246, 246, 6)
Conv2D | output: | (None, 242, 242, 6)
Layer_1_Pooling | input: | (None, 242, 242, 6)
AveragePooling2D | output: | (None, 121, 121, 6)
Layer_3_Conv | input: | (None, 121, 121, 6)
Conv2D | output: | (None, 117, 117, 10)
Layer_4_Conv | input: | (None, 117, 117, 10)
Conv2D | output: | (None, 113, 113, 10)
Layer_2_Pooling | input: | (None, 113, 113, 10)
AveragePooling2D | output: | (None, 56, 56, 10)
Layer_5_Conv | input: | (None, 56, 56, 10)
Conv2D | output: | (None, 52, 52, 20)
Layer_6_Conv | input: | (None, 52, 52, 20)
Conv2D | output: | (None, 48, 48, 20)
Layer_3_Pooling | input: | (None, 48, 48, 20)
AveragePooling2D | output: | (None, 24, 24, 20)
Layer_7_Conv | input: | (None, 24, 24, 20)
Conv2D | output: | (None, 20, 20, 20)
Layer_8_Conv | input: | (None, 20, 20, 20)
Conv2D | output: | (None, 16, 16, 20)
Layer_4_Pooling | input: | (None, 16, 16, 20)
AveragePooling2D | output: | (None, 8, 8, 20)
flatten | input: | (None, 8, 8, 20)
Flatten | output: | (None, 1280)
Layer_1_Dense | input: | (None, 1280)
Dense | output: | (None, 128)
Layer_2_dense | input: | (None, 128)
Dense | output: | (None, 4)

Fig. 6. Two-dimensional representation of network parameters and network architecture.



```
Epoch 20/20
105/105 [==============================] - 8s 81ms/step -
loss: 0.1162 - accuracy: 0.9585 - val_loss: 0.7878 - val_accuracy: 0.8083
```

Fig. 7. Metrics achieved in 20 epochs of training.

120 images each for the testing and validation process. The data augmentation process was performed on the 960 training images. The data augmentation was performed with TensorFlow. The transformations used were the change of image contrast, horizontal and vertical flips, and partial rotations.

Resizing was also performed, to obtain the same size for all data and to standardize the network inputs, the images were resized to 250×250 pixels. For each original image, 6 new images were generated by data augmentation. The number of generated images was chosen due to the limited amount of original images available for training and the limited computational power, resulting in a good balance that proved to be efficient considering the results, with the transformation applied either directly to the original image or to a previously generated image.

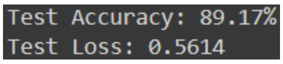
Fig. 2 shows that the original image belongs to the glioma class and Fig. 3 shows an example of the 6 transformations applied to that image, which, as mentioned previously, were a combination of changes in image contrast, horizontal and vertical flips, and partial rotations. A total of 6720 images were generated for model training, in which each class consisted of 1680 images. The final dimensions and sizes of the training, test, and validation sets are shown in Fig. 4.

3.3 Model Architecture

To reduce dimensionality, extract features and avoid overfitting, a convolutional network architecture with alternating convolutional layers and pooling layers was chosen. Fig. 5 shows a three-dimensional representation of the network architecture. The architecture of the model can be seen systematically in Fig. 6, which shows the layers that make up the CNN model. It consists of eight convolutional layers, four pooling layers, one fully connected layer, and one dense layer. Each pair of convolutional layers contains 6, 10, 20, and 20 kernels, all of size 5×5, and each layer also contains a ReLu activation function. In contrast, the size of the pooling layers is 2×2. The first convolutional layer receives input data of size 250×250. The dense layers at the end are used for classification into the four diagnostic categories.

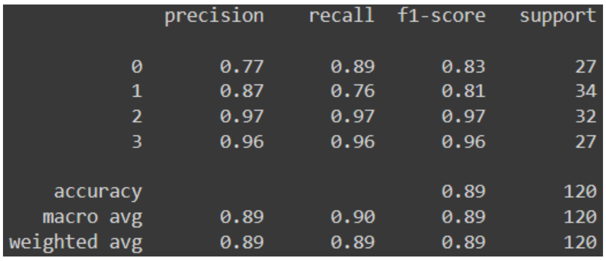
3.4 Model Compilation and Training

The model was created with the Adam optimizer and a learning rate of 0.001, using categorical cross entropy as the loss function. It was trained for 20 epochs with batches of 64 images, with hyperparameters adjusted based on performance during validation.



```
Test Accuracy: 89.17%
Test Loss: 0.5614
```

Fig. 8. Metrics achieved on testing.



```
              precision    recall  f1-score   support

           0       0.77      0.89      0.83        27
           1       0.87      0.76      0.81        34
           2       0.97      0.97      0.97        32
           3       0.96      0.96      0.96        27

    accuracy                           0.89       120
   macro avg       0.89      0.90      0.89       120
weighted avg       0.89      0.89      0.89       120
```

Fig. 9. Model metrics report.

3.5 Model Evaluation

The model was evaluated for accuracy and loss using the test set. A metrics report and confusion matrix were used to analyze performance by class and identify areas for improvement. The metrics report included precision, recall, and f1-score, the most commonly used metrics to evaluate deep learning models. In addition, the confusion matrix allows visualization of model performance, making it possible to identify which images were classified correctly and which were misclassified. This complements the metrics report.

4 Results

The results of the present study show that the convolutional neural network model has good accuracy in classifying magnetic resonance images for brain tumor detection. The most important results are listed below.

4.1 Dataset Dimensions

The model was trained using an augmented set of 6720 images, validated with 120 original images, and tested with another 120 original images. This data split allowed for correct training and fair evaluation of the model, despite the limited amount of data in the original set.

4.2 Training Evaluation

During the training of the model, significant progress was observed in the ability to correctly classify the images into the four defined categories. At the end of the 20 training epochs, a training accuracy of 95.85% and a loss of 0.1162 was achieved, as can be seen in Fig. 7. In contrast, the validation accuracy was 80.83% with a loss of 0.7878. These metrics indicate a possible slight overfitting of the model to the training set, possibly due to the limited amount of validation data combined with the data augmentation procedure during network training.

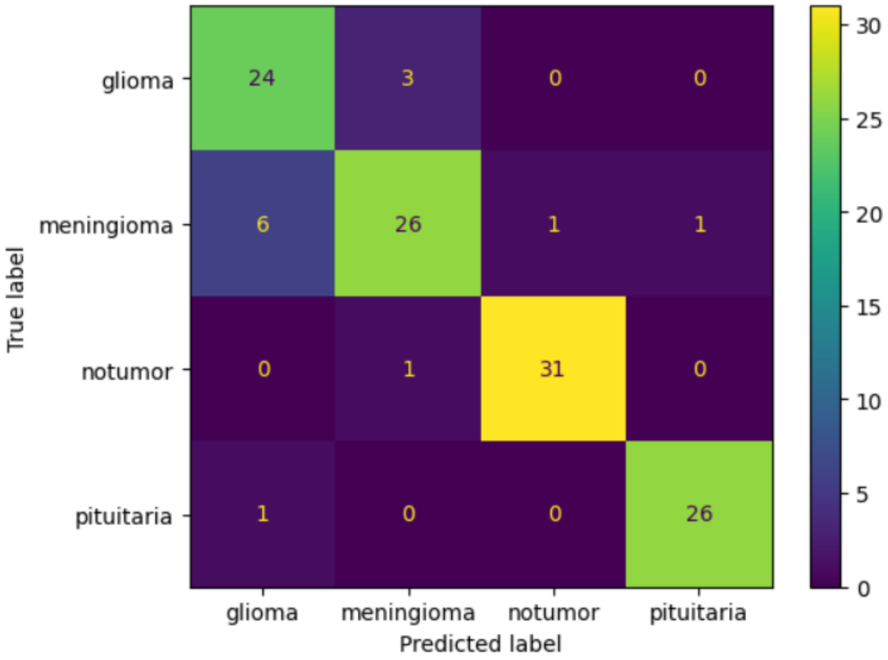


Fig. 10. Confusion matrix for the model.

4.3 Test Evaluation

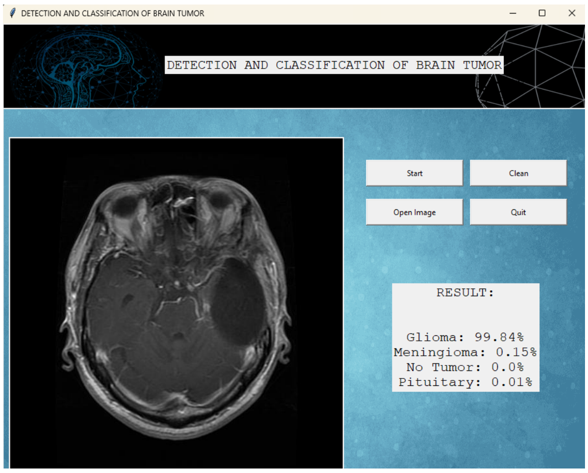
Fig. 8 shows that evaluation of the model on the test set yielded an accuracy of 89.17% and a loss of 0.5614, confirming the model's ability to generalize to novel images not seen during training. These results are particularly encouraging as efficient classification of brain tumors from magnetic resonance images is challenging due to the variability in tumor appearance.

4.4 Metrics Report

The analysis of the classification metrics shows a high precision, sensitivity (recall), and F1 score in all categories. As shown in Fig. 9, the No Tumor category performed particularly well with a recall of 0.97 and an F1 score of 0.97 (in the report the categories were labeled Glioma $=$ 0, Meningioma $=$ 1, No Tumor $=$ 2, and Pituitary $=$ 3), indicating a high true-positive rate and a good balance between precision and sensitivity. The other categories showed comparable results with values between 0.81 and 0.83 for the F1 score for glioma and meningioma and 0.96 for pituitary.

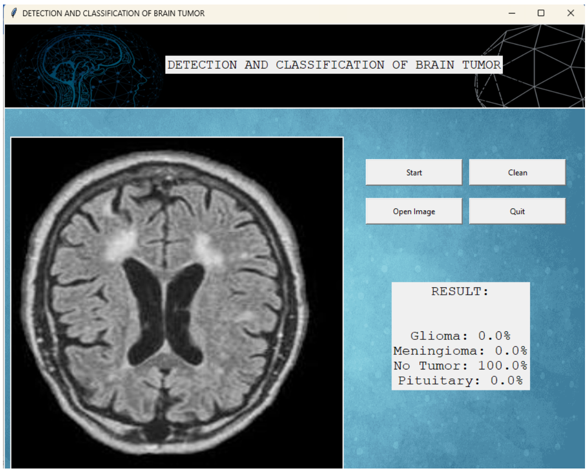
4.5 Confusion Matrix

As can be seen from the obtained confusion matrix, Fig. 10, the model showed a high ability to identify images without tumors, with remarkable accuracy in this category, being able to correctly identify 31 out of 32 cases. However, there were some


DETECTION AND CLASSIFICATION OF BRAIN TUMOR
DETECTION AND CLASSIFICATION OF BRAIN TUMOR
Start
Clean
Open Image
Quit

```
RESULT:

Glioma: 99.84%
Meningioma: 0.15%
No Tumor: 0.0%
Pituitary: 0.01%
```

Fig. 11. Prediction for a Glioma MRI image from the test set.


DETECTION AND CLASSIFICATION OF BRAIN TUMOR
DETECTION AND CLASSIFICATION OF BRAIN TUMOR
Start
Clean
Open Image
Quit

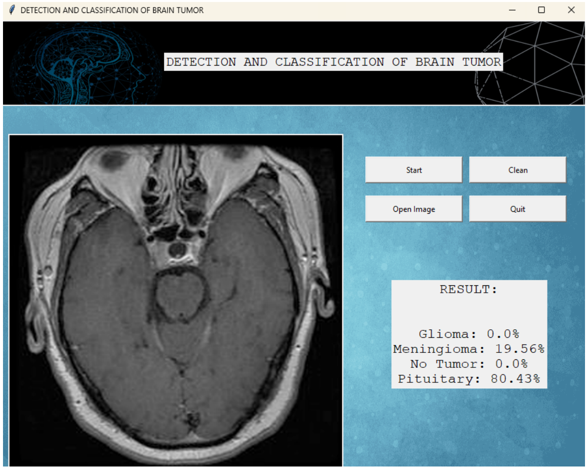
```
RESULT:

Glioma: 0.0%
Meningioma: 0.0%
No Tumor: 100.0%
Pituitary: 0.0%
```

Fig. 12. Prediction for a no tumor MRI image from the test set.

difficulties in accurate classification between tumor types, particularly between glioma and meningioma, where some cases of misclassification were observed.

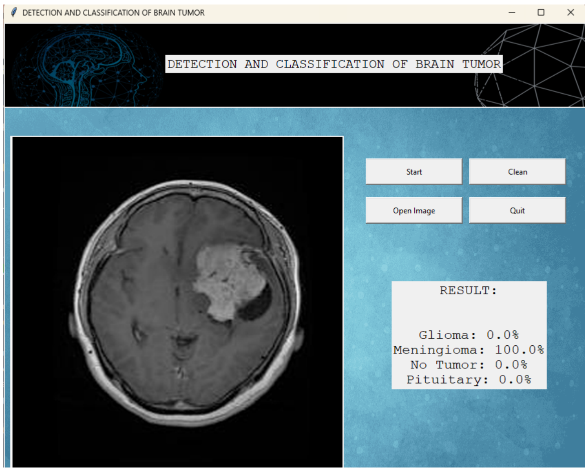
4.6 Implementation in an Executable Program Using a Graphical Python Interface

The model was implemented in a graphical Python interface by creating an executable file. This was done to allow any user to interact with and use the model without the need to have Python installed on their system and without the need to have technical knowledge of how to use the model. The graphical interface created is quite intuitive, so it can be used by users who are not at all familiar with managing deep learning prediction models. It is based on the Python library tkinter and allows loading MRI


DETECTION AND CLASSIFICATION OF BRAIN TUMOR
DETECTION AND CLASSIFICATION OF BRAIN TUMOR
Start
Clean
Open Image
Quit

```
RESULT:

Glioma: 0.0%
Meningioma: 19.56%
No Tumor: 0.0%
Pituitary: 80.43%
```

Fig. 13. Prediction for a Pituitary MRI image from [5].


DETECTION AND CLASSIFICATION OF BRAIN TUMOR
DETECTION AND CLASSIFICATION OF BRAIN TUMOR
Start
Clean
Open Image
Quit

```
RESULT:

Glioma: 0.0%
Meningioma: 100.0%
No Tumor: 0.0%
Pituitary: 0.0%
```

Fig. 14. Prediction for a Meningioma MRI image from [5].

images for later analysis (via the “Open image” button). As a result (after processing and evaluation by the model, if you click on the “Start” button”) the loaded image and the membership probabilities predicted by the model for each of the 4 possible classes are displayed. There is also a button to “Clear” the interface, which deletes the results of the last analysis and leaves the interface clean and ready to load a new image, and the “Exit” button, which allows you to close the program at any time. Below is an example of what the graphical user interface looks like after analyzing brain tumor images in the implementation. Figs. 11-12 show the results of the classification of two images from the test set, while Figs. 13-14 are predictions of MRIs from a different dataset [5], showing successful results as well.

5 Conclusions

This study has demonstrated that the use of deep learning techniques, specifically Convolutional Neural Networks (CNNs), together with data augmentation methods, can be of great utility for the detection and classification of brain tumors in Magnetic Resonance Images. The network architecture designed for this research has proven to be effective in classifying gliomas, meningiomas, pituitary tumors, and non-tumors, achieving an overall test set accuracy of 89.17%.

The results obtained suggest several important conclusions. First of all, high performance in the detection of non-tumors (healthy patients) was seen: the model demonstrated an outstanding ability to identify non-tumor images, which is key to preventing false positive diagnoses in clinical settings. Additionally, challenges in tumor type classification were observed: although the model has shown high overall accuracy, it still faces challenges in differentiating between certain tumor types, such as glioma and meningioma. This could indicate a need for further adjustments to the model architecture or training approach to improve specificity or a need for a larger amount of training data and more data for each class.

Likewise, data augmentation proved to be essential in improving the model's ability to generalize to new images, as initially the data set available was notably limited, and having trained with this set would likely have led to inferior results due to the restricted number of examples in each class. This approach could be further explored to include other augmentation techniques that could help improve the distinction between similar classes. On the other hand, the difference between training and validation accuracy suggests overfitting, a more or less expected result due to the limited validation set. Strategies such as regularization, dropout, adding callbacks to the training process, or increasing the validation set could be investigated to mitigate this effect.

Finally, the fact of having created a Python graphical interface compiled as an executable file contributed greatly to making the use of the model easier and more intuitive, and added value to the final work. In this way, the use of this type of predictive model is brought closer to the common user, who does not necessarily need to know the strategies that must be followed normally to be able to use such models correctly, for instance, when following a Jupyter notebook or executing a Python script.

Future work could explore the optimization of the network architecture, fine-tuning of hyperparameters from larger pre-trained networks, and expansion of the data set to improve both the training process and the validation and evaluation process (possibly by collaborating with medical institutions to acquire more images and getting in touch with specialists from these institutions to start developing a way to utilize the graphical interface to assist in making appropriate diagnoses). In turn, the incorporation of other data modalities, such as clinical patient data, could help improve the accuracy and robustness of the model.

Acknowledgments. The authors would like to thank the Mexican Council of Science and Technology for the scholarship they received, and Dr. Sebastián Salazar-Colores for encouraging them to write it and submit it to contest. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Al-Zoghby, A. M., Al-Awadly, E. M. K., Moawad, A., Yehia, N., Ebada, A. I.: Dual deep cnn for tumor brain classification. Diagnostics, vol. 13, no. 12, pp. 2050 (2023) doi: 10.3390/diagnostics13122050
2. Amin, J., Sharif, M., Gul, N., Raza, M., Anjum, M. A., Nisar, M. W., Bukhari, S. A. C.: Brain tumor detection by using stacked autoencoders in deep learning. Journal of Medical Systems, vol. 44, no. 32 (2019) doi: 10.1007/s10916-019-1483-2
3. Bakas, S., Reyes, M., Jakab, A., Bauer, S., Rempfler, M., Crimi, A., Shinohara, R. T., Berger, C., Ha, S. M., Rozycki, M., Prastawa, M., Alberts, E., Lipková, J., Freymann, J. B., Kirby, J. S., Bilello, M., Fathallah-Shaykh, H. M., Wiest, R., Kirschke, J., Wiestler, B., et. al.: Identifying the best machine learning algorithms for brain tumor segmentation, progression assessment, and overall survival prediction in the BRATS challenge (2018) doi: 10.48550/arXiv.1811.02629
4. Bhanothu, Y., Kamalakannan, A., Rajamanickam, G.: Detection and classification of brain tumor in MRI images using deep convolutional network. In: Proceedings of the 6th International Conference on Advanced Computing and Communication Systems, pp. 248–252 (2020) doi: 10.1109/icaccs48705.2020.9074375
5. Cheng, J.: Brain tumor dataset. Figshare (2024) doi: 10.6084/m9.figshare.1512427.v57
6. Das, S., Riaz-Rahman-Aranya, O. F. M., Labiba, N. N.: Brain tumor classification using convolutional neural network. In: Proceedings of the 1st International Conference on Advances in Science, Engineering and Robotics Technology , pp. 1–5 (2019) doi:10.1109/icasert.2019.8934603
7. Gupta, S.: Brain MRI Scans for brain tumor classification (2023) www.kaggle.com/datasets/shreyag1103/brain-mri-scans-for-brain-tumor-classification/data
8. Han, C., Rundo, L., Araki, R., Furukawa, Y., Mauri, G., Nakayama, H., Hayashi, H.: Infinite brain mr images: PGGAN-based data augmentation for tumor detection. Neural Approaches to Dynamics of Signal Exchanges. Smart Innovation, Systems and Technologies, vol. 151, pp. 291–303 (2019) doi: 10.1007/978-981-13-8950-4_27
9. Menze, B. H., Jakab, A., Bauer, S., Kalpathy-Cramer, J., Farahani, K., Kirby, J., Burren, Y., Porz, N., Slotboom, J., Wiest, R., Lanczi, L., Gerstner, E., Weber, M., Arbel, T., Avants, B. B., Ayache, N., Buendia, P., Collins, D. L., Cordier, N., Corso, J. J., et. al.: The multimodal brain tumor image segmentation benchmark (brats). IEEE Transactions on Medical Imaging, vol. 34, no. 10, pp. 1993–2024 (2015) doi: 10.1109/tmi.2014.2377694
10. Mohsen, H., El-Dahshan, E. A., El-Horbaty, E. M., Salem, A. M.: Classification using deep learning neural networks for brain tumors. Future Computing and Informatics Journal, vol. 3, no. 1, pp. 68–71 (2018) doi: 10.1016/j.fcij.2017.12.001
11. Nazir, M., Shakil, S., Khurshid, K.: Role of deep learning in brain tumor detection and classification (2015 to 2020): A review. Computerized Medical Imaging and Graphics, vol. 91, pp. 101940 (2021) doi: 10.1016/j.compmedimag.2021.101940
12. Özkaraca, O., Bağrıaçık, O. İ., Gürüler, H., Khan, F., Hussain, J., Khan, J., Laila, U. E.: Multiple brain tumor classification with dense CNN architecture using brain MRI images. Life, vol. 13, no. 2, pp. 349 (2023) doi: 10.3390/life13020349
13. Rai, H. M., Chatterjee, K.: Detection of brain abnormality by a novel Lu-Net deep neural CNN model from mr images. Machine Learning with Applications, vol. 2, pp. 100004 (2020) doi: 10.1016/j.mlwa.2020.100004
14. Srinivasan, S., Francis, D., Mathivanan, S. K., Rajadurai, H., Shivahare, B. D., Shah, M. A.: A hybrid deep CNN model for brain tumor image multi-classification. BMC Medical Imaging, vol. 24, no. 1 (2024) doi: 10.1186/s12880-024-01195-7

15. Ucuzal, H., YASAR, S., Colak, C.: Classification of brain tumor types by deep learning with convolutional neural network on magnetic resonance images using a developed web-based interface. In: Proceedings of the 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies, pp. 1–5 (2019) doi: 10.1109/ismsit.2019.8932761
16. Vankdothu, R., Hameed, M. A., Fatima, H.: A brain tumor identification and classification using deep learning based on cnn-lstm method. Computers and Electrical Engineering, vol. 101, pp. 107960 (2022) doi: 10.1016/j.compeleceng.2022.107960