

An Adapted Convolutional Neural Network for Brain Tumor Detection

Kamagaté Beman Hamidja¹, Kanga Koffi¹, Brou Pacôme¹, Olivier Asseu^{1,2}, Souleymane Oumtanaga³

¹Ecole Supérieure Africaine des Technologies de l'Information et de la Communication (ESATIC), LASTIC, Abidjan, Côte d'Ivoire

²UMRI STI, INPHB INP—Houphouët Boigny, Yamoussoukro, Côte d'Ivoire

³UMRI MSN, INPHB INP—Houphouët Boigny, Yamoussoukro, Côte d'Ivoire

Email: beman.kamagate@esatic.edu.ci

How to cite this paper: Hamidja, K.B., Koffi, K., Pacôme, B., Asseu, O. and Oumtanaga, S. (2024) An Adapted Convolutional Neural Network for Brain Tumor Detection. *Open Journal of Applied Sciences*, 14, 2809-2825.
<https://doi.org/10.4236/ojapps.2024.1410184>

Received: September 17, 2024

Accepted: October 20, 2024

Published: October 23, 2024

Copyright © 2024 by author(s) and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

In medical imaging, particularly for analyzing brain tumor MRIs, the expertise of skilled neurosurgeons or radiologists is often essential. However, many developing countries face a significant shortage of these specialists, which impedes the accurate identification and analysis of tumors. This shortage exacerbates the challenge of delivering precise and timely diagnoses and delays the production of comprehensive MRI reports. Such delays can critically affect treatment outcomes, especially for conditions requiring immediate intervention, potentially leading to higher mortality rates. In this study, we introduced an adapted convolutional neural network designed to automate brain tumor diagnosis. Our model features fewer layers, each optimized with carefully selected hyperparameters. As a result, it significantly reduced both execution time and memory usage compared to other models. Specifically, its execution time was 10 times shorter than that of the referenced models, and its memory consumption was 3 times lower than that of ResNet. In terms of accuracy, our model outperformed all other architectures presented in the study, except for ResNet, which showed similar performance with an accuracy of around 90%.

Keywords

Brain Tumor, MRI, Convolutional Neural Network, KKDNet, GoogLeNet, DensNet, ResNet, ShuffleNet

1. Introduction

The human nervous system is extremely complex and plays a crucial role in

processing and transmitting neural signals that govern both voluntary and involuntary functions throughout the body, with the brain as its primary component. The brain governs most bodily functions, taking charge of analyzing, integrating, organizing, decision-making, and directing the rest of the body [1]. The anatomical structure of the brain is remarkably complex [2]. It consists of various distinct yet interconnected structures, such as the cerebral cortex, the cerebellum, and the brainstem. Each of these regions plays a crucial role in processing information and regulating bodily functions.

Some brain disorders, such as strokes, infections, headaches and brain tumors, are particularly challenging to diagnose, analyze, and treat appropriately. A brain tumor is a mass of abnormal cells that develops within the rigid skull enclosing the brain. Its expansion in such a confined area can cause issues, such as brain damage, which poses a serious risk to the brain [3]. There are 130 types of tumors that can affect the brain and central nervous system (CNS), ranging from benign to malignant and from extremely rare to relatively common.

These tumors are systematically classified into primary and secondary categories [4]. Primary brain tumors originate directly within the brain, either forming from brain cells or being encased by surrounding nerve cells. These tumors can be benign or malignant [5]. Secondary brain tumors, which account for the majority of brain malignancies, are often fatal. These tumors develop when cancers, such as those in the breast, kidney, or skin, spread silently from another part of the body to the brain. Unlike benign tumors, these malignant are notorious for their relentless ability to spread to other parts of the body [6].

According to research [7], brain tumors account for approximately 85% to 90% of significant central nervous system (CNS) tumors. Early detection is crucial for significantly reducing the mortality rate associated with these tumors. Healthcare professionals commonly use magnetic resonance imaging (MRI) for early diagnosis [8]. Although MRI scans are highly effective for diagnosing brain tumors, they have several limitations that can reduce their effectiveness.

In medical imaging, particularly MRI analysis, the expertise of a professional neurosurgeon or radiologist is often crucial. However, many developing countries face a significant shortage of neurosurgeons and radiologists with the specialized skills needed to accurately identify and analyze tumors. This shortage intensifies the difficulty of delivering precise and timely diagnoses and lengthens the process of producing comprehensive MRI reports. Such delays in patient care can have serious implications for treatment outcomes, particularly for conditions requiring urgent attention, ultimately leading to increased mortality rates [9].

Additionally, the lack of access to continuous medical education and modern diagnostic tools in these regions further complicates the issue. Without a robust infrastructure to support the learning and dissemination of advanced medical techniques, many healthcare professionals in developing countries are not equipped with the skills necessary to interpret complex MRI results effectively. This knowledge gap not only impairs the accurate diagnosis of tumors but also

diminishes the overall quality of healthcare services available to patients [10].

To overcome these challenges, implementing an automated system on a cloud-based platform offers a promising solution [11]. Such a system could utilize artificial intelligence and machine learning algorithms [12] to analyze MRI scans with high accuracy, reducing the reliance on limited medical expertise. By automating the initial stages of MRI analysis, healthcare providers in developing countries could drastically reduce the time required to produce diagnostic reports. This would not only speed up the diagnostic and treatment process but also improve the overall efficiency of healthcare systems in these regions.

In this work, we developed a framework based on an adapted novel Convolutional Neural Network (CNN) architecture to classify types of brain tumors: glioma, meningioma, pituitary tumors, and no tumor. The main contribution is that the proposed CNN is designed to be less deep while achieving similar accuracy, with reduced execution time and lower memory consumption compared to some of the complex CNN architectures found in the literature.

The remainder of this paper is organized as follow. Section 2 describes the related work. Section 3 covers the proposed architecture, its optimization, the data models, and the framework for brain tumor classification. Section 4 presents the results and discussion. We conclude the paper with Section 5.

2. Related Work

This section provides a detailed examination of studies utilizing machine learning and deep learning models for classifying brain tumors. It explores the various approaches and methodologies employed in these studies, highlighting the advancements achieved and the challenges faced with each model.

Song *et al.* [13] employed numerous machine learning algorithms to classify MRI brain tumor images. Specifically, five models were tested, including logistic regression, Stochastic Gradient Descent, decision tree, Support Vector Machine (SVM), and k-Nearest Neighbors (k-NN). The dataset used consists of four categories of MRI images: normal tissue, glioma, meningioma, and pituitary tumors. To improve efficiency and reduce execution time, a random selection of 250 images was used instead of the entire dataset. After importing these 250 images, they were resized and subjected to dimensionality reduction using Principal Component Analysis (PCA). The data was then split into training and testing sets, with 80% of the data used for training. Several machine learning models were trained using the training data, and their effectiveness was assessed on the testing dataset to determine their performance in real-world scenarios. The results revealed that k-Nearest Neighbors (k-NN) outperformed the other models in terms of efficiency.

Ankit *et al.* [14] undertook a similar task using a brain MRI image dataset, initially focusing on binary classification to determine the presence or absence of a tumor with nine different machine learning algorithms. These algorithms were evaluated and compared using various metrics, including accuracy, recall,

precision, and F1-score. Among them, the Gradient Boosting classifier proved to be the most effective, achieving a recall of 94.4%, an accuracy of 92.4%, an F1-score of 89.5%, and a precision of 85%. To tackle the multi-class classification problem, four machine learning algorithms which include SVM, KNN, Random Forest, and XGBoost were employed. The dataset included four types of brain MRI images: glioma tumor, meningioma tumor, pituitary tumor, and no tumor. The performance of these algorithms was evaluated based on accuracy, precision, recall, and F1-score. XGBoost outperformed the other algorithms across all metrics, achieving 90% in accuracy, precision, recall, and F1-score.

In addition to the previously mentioned studies, many other research efforts utilize machine learning algorithms for brain tumor classification. A significant review is provided in work [15]. These algorithms have demonstrated promising performance while remaining relatively simple. However, they require substantial preprocessing of MRI images before application. Moreover, these models may struggle to detect complex relationships within the data, which is crucial for brain tumors, especially at an early stage [15]. Early diagnosis and treatment of brain tumors have the potential to decrease death rates.

Recent studies utilizing Convolutional Neural Networks (CNNs) [16] have demonstrated their ability to work directly with raw images and capture intricate relationships and subtle features. This ability frequently leads to enhanced accuracy in brain tumor detection, especially in complex cases, at early stages, and with large datasets.

Mahmud *et al.* [17] propose a CNN architecture for the efficient identification of brain tumors using MR images. They evaluate various models, including ResNet-50, VGG16, and InceptionV3 (GoogLeNet), and perform a comparative analysis based on performance metrics such as accuracy, recall, and loss, using a dataset of 3264 MRI images. The accuracy of ResNet-50, VGG16, and InceptionV3 were 81.80%, 71.60%, and 80.00%, respectively. For recall, the performance values were 81.04%, 70.03%, and 79.81%, respectively. The loss metrics were 0.85%, 1.85%, and 3.67%, respectively. The proposed CNN model achieved an accuracy of 90.3%, a recall of 91.19%, and a loss of 0.25, outperforming ResNet, VGG16, and InceptionV3 in the early detection of various brain tumors.

Kameswara *et al.* [18] introduced a modified U-Net architecture based on residual networks. Originally designed for medical image segmentation, the U-Net model is enhanced with periodic reorganization in the encoder phase and sub-pixel convolution in the decoder phase. This revised U-Net model was assessed on two datasets, achieving segmentation accuracies of 93.40% and 92.20%, respectively. Beyond the previously mentioned studies, several other works [19]-[21] utilize specific CNN architectures, often through transfer learning, to classify brain tumors from MRI images. While these models offer notable advancements in detecting complex data interactions and early-stage tumors, their complexity results in increased computational time and memory usage. To overcome these limitations, we propose a new CNN model in the following section.

3. Methodology

In very deep Convolutional Neural Network (CNN), gradients can become extremely small or excessively large during backpropagation, making training difficult, slow, or even unstable. A deeper network also requires more computations, which lengthens the training time and increases the resources needed (CPU, GPU). Each additional layer introduces new parameters, thereby increasing the memory requirements for storing the model and the intermediate data during training. Furthermore, an excessive number of layers can lead to overfitting to the specific details of the training data, which may compromise performance on new data (poor generalization) [16]. Additionally, as the network becomes deeper, it becomes increasingly difficult to understand how it makes decisions, making the interpretation of the models more complex [22].

Considering the aforementioned drawbacks, we introduce an adapted CNN architecture that prioritizes robustness without depending on layer depth. Our approach focuses on the quality of feature extraction and ensures model stability across diverse scenarios. By employing a shallower architecture, we aim to reduce model complexity and minimize the risk of overfitting, while also achieving faster training convergence. Additionally, by concentrating on task-relevant and meaningful features, our method seeks to enhance generalization efficiency, even with a limited number of layers.

3.1. Convolutional Neural Networks Layers

Convolutional Neural Networks (CNNs) are a class of neural networks primarily used for analyzing, recognizing, and classifying images [23]. It comprises an input layer, several hidden layers, and an output layer (see **Figure 1**).

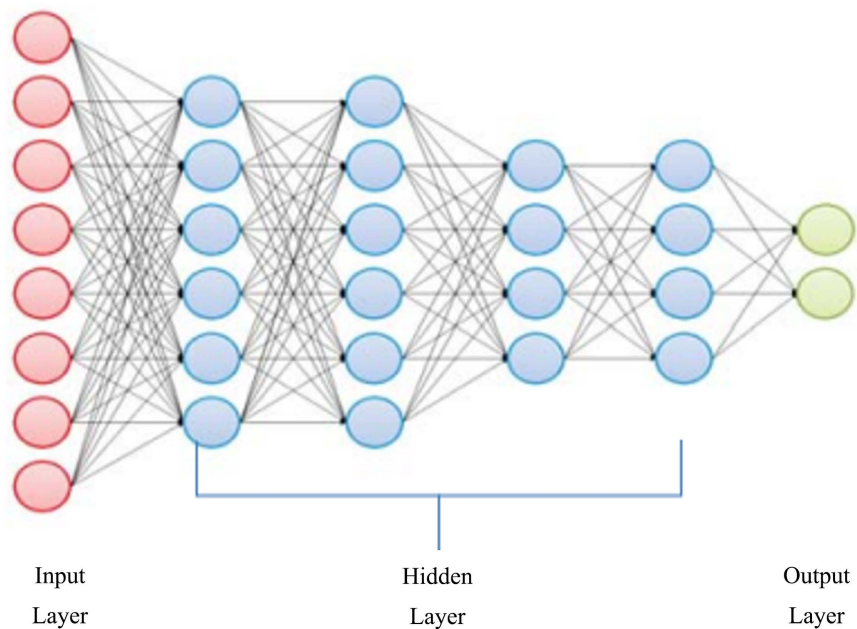


Figure 1. General architecture of Convolutional Neural Network (CNN) [24].

The input layer is responsible for receiving raw data, such as images, and passing it into the network.

The hidden layers include the convolutional, pooling, activation, and fully connected layers, each fulfilling a distinct role. They are the essential layers [25], collaborating to enable CNNs to learn and identify patterns in data, making these networks highly effective for tasks like image recognition, object detection, and more.

The convolution layer detects features by applying filters (kernels) to the input data to extract local patterns like edges, textures, or shapes from images. It performs convolution operations, a mathematical process that combines two signals to produce a third one [26]. This operation measures the effect of one signal on another, influenced by the shape of the second signal. The computation is outlined below.

$$(p \times q)[s] = \sum_{r=-\infty}^{\infty} p[r]q[s-r] \quad (1)$$

where, p and q are functions that can be either discrete or continuous, and s is the position of the output signal. For discrete signals, the equation is adjusted to

$$(p \times q)[s] = \sum_{r=-\infty}^{\infty} p[r]q[s-r]\Delta r \quad (2)$$

with Δr the sampling interval. For continuous one, the equation takes the form:

$$(p \times q)[u] = \int_{-\infty}^{\infty} p(\sigma)q(u-\sigma)d\sigma \quad (3)$$

Here, u denotes the position of the output signal.

The pooling layer, placed between convolutional layers is intended to reduce the spatial dimensions (width and height) of the feature maps produced by the convolutional layer [23]. It enhances the robustness of feature extraction and decreases the risk of overfitting. Generally used pooling operations include average pooling, max pooling and global pooling [27].

To enable the model to capture complex and non-linear patterns found in real-world scenarios, an activation layer is necessary, which performs activation functions. Common activation functions found in the literature include the Sigmoid activation function, the Hyperbolic Tangent (Tanh) activation function, and the Rectified Linear Unit (ReLU) activation function. ReLU is commonly used in hidden layers due to its simplicity and efficiency [28]. Several variations of the ReLU activation function have been proposed to enhance its performance, such as the Leaky ReLU and Parametric ReLU (PReLU) activation functions.

The normal ReLU neutralizes negative values while preserving positive values unchanged, a Leaky ReLU is variant that allows a small, non-zero gradient when the input is negative. It is computing as follow:

$$\text{Leaky ReLU}(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha x & \text{if } x \leq 0 \end{cases} \quad (4)$$

where α is a small constant for example $\alpha = 0.001$. x is range from $-\infty$ to

$+\infty$. PReLU is similar to Leaky ReLU but α a learnable slope for negatives values.

The fully connected layer, used in the final stages to generate the output, is a layer where each neuron is connected to every neuron in the previous layer [29]. it is in charge of integrating features extracted from input data by previous layers. Fully connected layers can learn non-linear combinations of input features, allowing them to capture complex relationships between these features. Once the features are effectively combined, the fully connected layer produces outputs used for classification in classification tasks or for predicting continuous values in regression tasks.

3.2. Proposed Convolutional Neural Network Architecture

To develop an optimal architecture, we tested various configurations by adjusting the number of layers, filter sizes, stride, and activation functions. We then selected the parameter values that led to the highest accuracy. The results obtained are presented in **Table 1**.

Table 1. Different CNN architecture.

Parameter	Tested value	Recommendation
Convolution layers	3, 4, 5, 6, 7, 8	4
Kernel	(3×3) , (5×5) , (7×7)	(3×3)
Stride	1, 2	2
Activation function	ReLU, Sigmoid, Softmax, Tanh	ReLU

As shown in **Table 1**, our proposal includes four convolutional layers, each producing an output that applies the ReLU activation function, followed by a pooling layer with a 2×2 filter and a stride of 2. The first convolutional layer uses 16 filters, each with a 3×3 pixel matrix. After this, a pooling layer with a 2×2 filter and a stride of 2 is applied. The second convolutional layer uses 32 filters, the third layer uses 64 filters, and the final layer uses 128 filters. **Figure 2**, the convolution part, shows that the four convolutional layers are preceded by the input layer and followed by the fully connected layer.

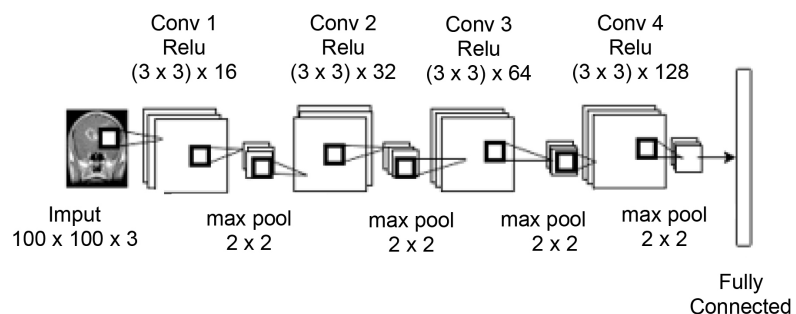


Figure 2. Convolution part.

To enhance the efficiency of feature discrimination in the convolutional phases, the fully connected section is trained using two distinct sub-networks (see **Figure 3**). The first subnetwork has 1024 neurons as input, followed by 512, 256 and 128 neurons respectively in the next three layers. The second subnetwork has 512 neurons as input, followed by 128 neurons in each of the three successive layers. Both subnetworks have an identical number of fully connected layers, and their outputs are concatenated in the fourth layer to extract the most discriminative features. This approach allows for the exploration of high-dimensional features using two distinct architectures. Each fully connected layer carries out complex computations to transform inputs into meaningful outputs, enhancing the model's ability to learn and generalize from the training data. The final layer is a SoftMax loss layer with 128 inputs.

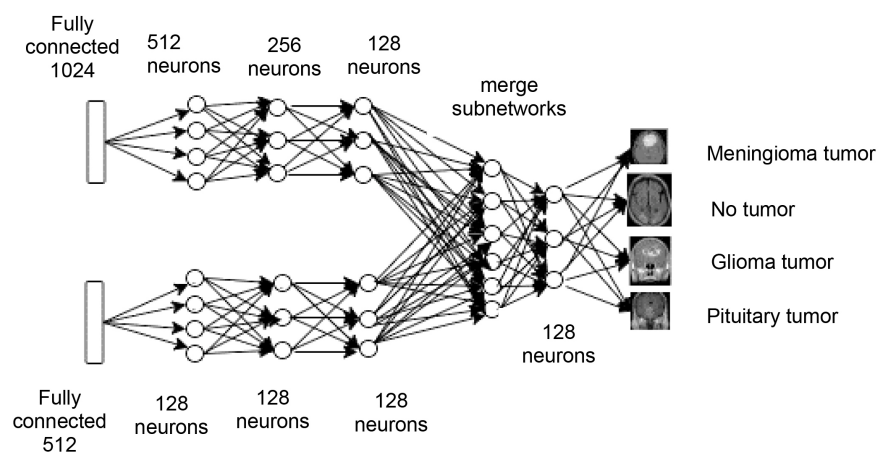


Figure 3. Fully connected part.

We optimize the weights of convolutional neural networks using the gradient descent algorithm. This iterative process fine-tunes key parameters, such as the base learning rate. We employed the cyclic learning rate method [30], which involves varying the learning rate within a range of values over a specific number of iterations. Adadelta optimizer was applied with a learning rate ranging from $1 \cdot e^{-6}$ to $1 \cdot e^{-1}$. As recommended by [31], we optimize the learning rate and batch size by experimenting with different values while keeping other hyperparameters constant. The specified hyperparameter space can then be explored further through random sampling, allowing us to select the parameters that yield the best results on the validation set. Following this approach, we have listed the other hyperparameter values used in the network in **Table 2**.

Table 2. Optimization hyperparameter.

Hyperparameter	Tested value	Recommendation
Epoch	16, 25, 32, 64, 50	25
Batch-size	25, 50, 75, 100	50

Continued

Per-parameter adaptive learning rate methods	SGD, RMSprop, Adagrad, Adadelta, Adam, Adamax, Nadam	Adadelta
Dropout rate	0.1, 0.2, 0.3, 0.4	0.1

3.3. Data Collection

In this study, we used publicly available data from kaggle.com [32], which consists of 3264 magnetic resonance imaging (MRI) images of brain tumors. The images are categorized into four classes: meningioma tumor, pituitary tumor, glioma tumor, and no tumor. **Table 3** presents a detailed partitioning of the dataset.

Table 3. Dataset partitioning.

Type of brain tumor	Quantity
Pituitary tumor	901
Meningioma Tumor	937
Glioma Tumor	926
Total	3264

Figure 4 (inspired from [17]) illustrates brain magnetic resonance images categorized by the different types of tumors.

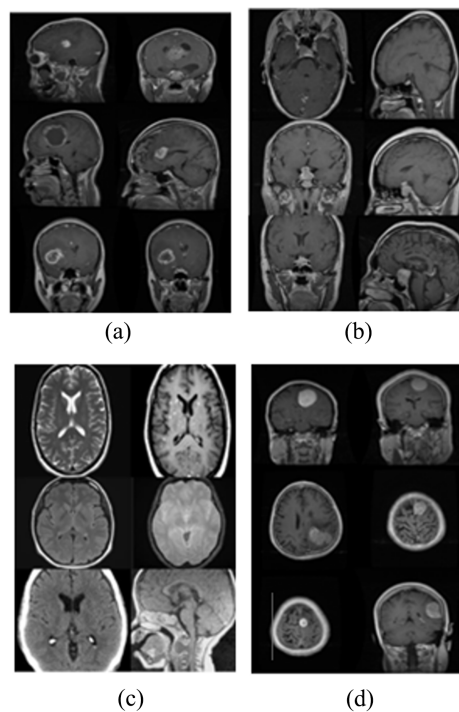


Figure 4. Images of brain tumors. (a) Glioma tumor; (b) Pituitary tumor; (c) No tumor; (d) Meningioma tumor.

3.4. Proposal Framework

To predict brain tumors, we follow these steps. First, we collect the data. Next, we conduct a data pre-processing phase, where we apply various techniques to eliminate noise and improve data quality. The second step involves splitting the data into training and validation datasets. 80% of the images from each type of brain tumor listed in **Table 2** are used for training, while the remaining images are used for validation. The third step consist of applying deep learning model based on our proposed CNN architecture with training dataset to perform brain tumor classification followed by the using of validation dataset. For comparison purpose we also use another CNN architecture like GoogLeNet, ShuffleNet, DenseNet and ResNet. For practical reasons, we decided to name our own proposed CNN architecture KKDNet. The output determines whether the input image is categorized as a meningioma tumor, pituitary tumor, glioma tumor, or no tumor. The final step of our framework involves evaluating performance by calculating accuracy, recall, AUC, and loss. **Figure 5** illustrates the processing steps of the proposed framework.

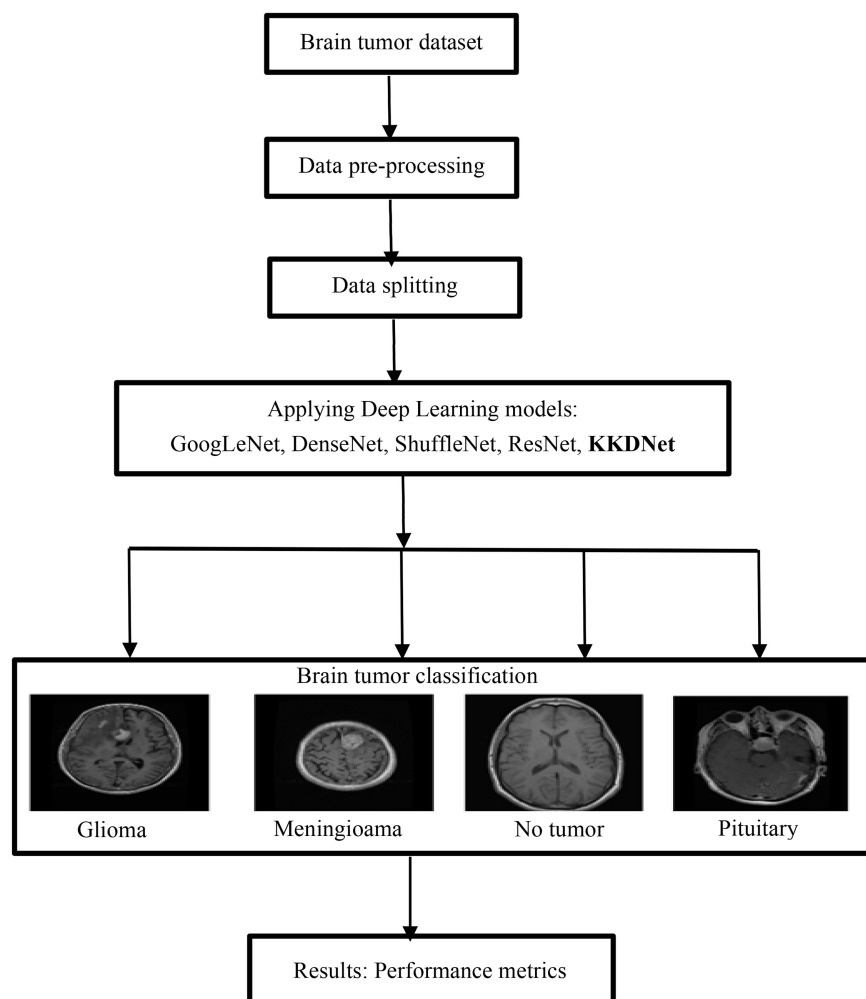


Figure 5. Brain tumor classification diagram.

4. Result and Discussion

In this section, we present the results obtained by conducting experiments on a Core i7 computer with 16 GB of RAM, using Python packages such as Scikit-learn and Keras. We implemented our proposed CNN architecture (KKDNet) as well as other architectures, including GoogLeNet, ShuffleNet, ResNet, and DenseNet. The metrics we evaluated include accuracy, execution time, and memory usage.

4.1. Results

Figure 6 shows the training and validation loss. We can observe that the training loss curve and validation loss curve decrease over the epochs. **Figure 7** presents the training and validation curves of the accuracy over the epochs. Training curve. The curve reaches a fairly high accuracy value quickly; by epoch 15, we surpass 90% accuracy. However, it is noted that the validation curve does not achieve equivalent performance in terms of accuracy. Until epoch 25, it remains below 80%.

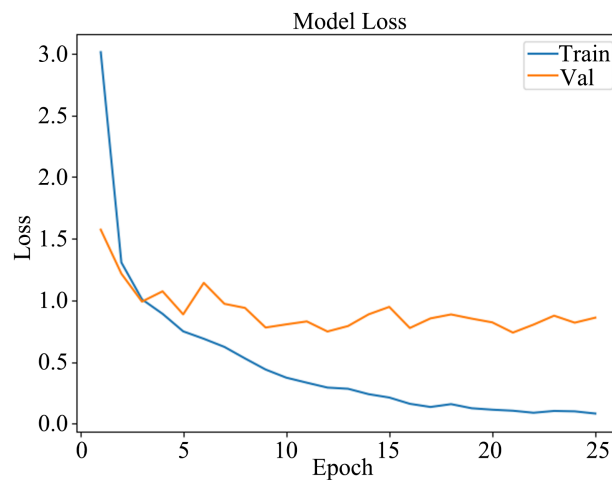


Figure 6. Training and validation loss curve of KKDNet.

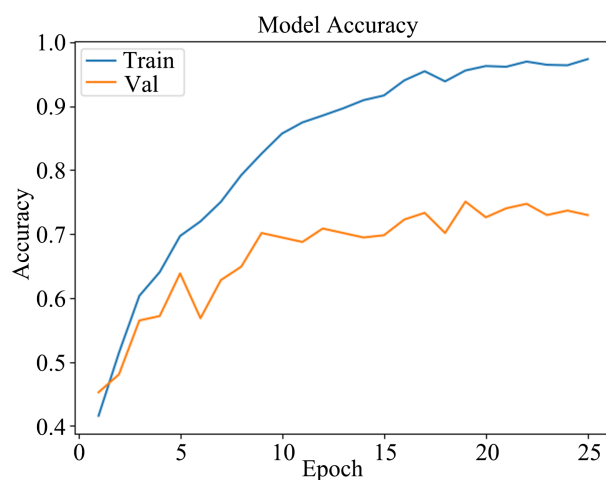


Figure 7. Training and validation accuracy curve of KKDNet.

To evaluate our model’s performance in terms of execution time and memory consumption, we represented, in **Figure 8**, the execution time in seconds of several models, including ours (KKDNet), in the form of a histogram. We observe that KKDNet achieves the best execution time compared to the other models, with an approximate time of around 500 seconds. The model closest in terms of execution time, GoogLeNet, has a time of about 700 seconds. In contrast, the model with the highest execution time reaches approximately 17,000 seconds, which is about 34 times the time required for our model (KKDNet).

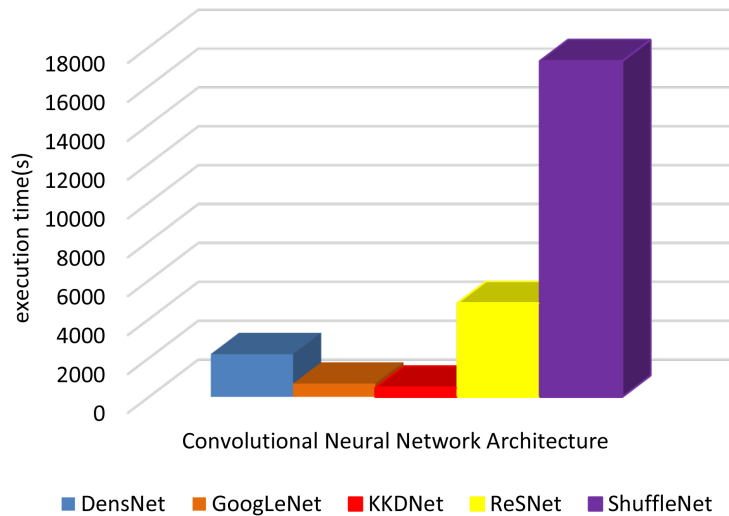


Figure 8. Execution time (s) of DensNet, GoogLeNet, KKDNet, ResNet and ShuffleNet.

The pie chart in **Figure 9** represents the memory consumption of different architectures. We observe that the memory requirements for ShuffleNet, DenseNet, and GoogLeNet are lower than those of our model. In contrast, ResNet’s memory consumption is more than three times higher than that of our model.

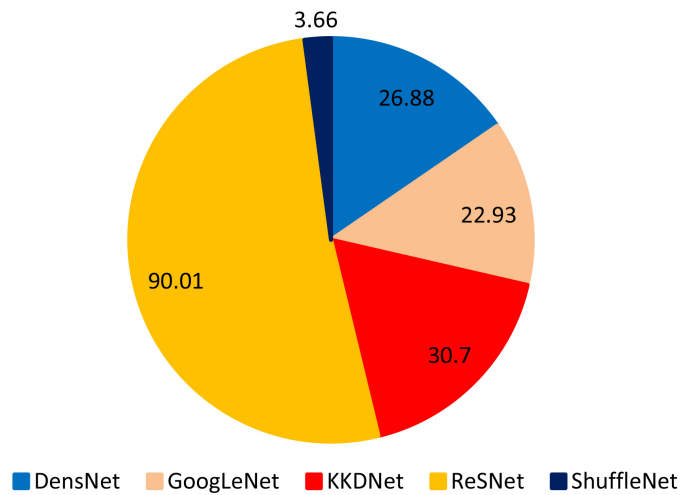


Figure 9. Amount of memory (MB) of DensNet, GoogLeNet, KKDNet, ResNet and ShuffleNet.

The final representation (**Figure 10**) is a histogram illustrating the performance of different architectures based on the accuracy metric. Our model, KKDNet, ranks among the top two, achieving performance levels comparable to ResNet, with an accuracy exceeding 90%.

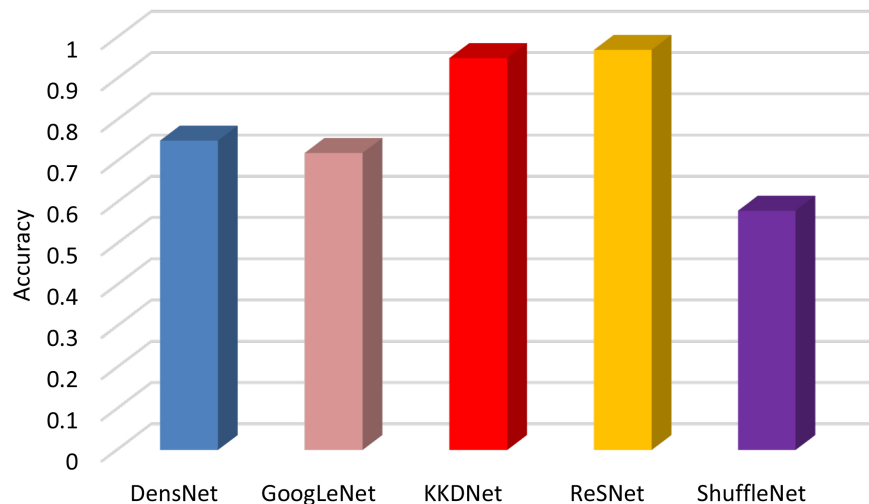


Figure 10. Accuracy of DensNet, GoogLeNet, KKDNet, ResNet and ShuffleNet.

4.2. Discussion

Figure 6 shows the training and validation loss. We can observe that the training loss curve decreases over the epochs, indicating that the model is gradually improving during training. The validation loss curve also decreases over the epochs, which suggests that the model is not experiencing overfitting and may continue to improve if the number of epochs is increased.

In **Figure 8**, the curve reaches a fairly high accuracy value quickly; by epoch 15, we surpass 90% accuracy. However, it is noted that the validation curve does not achieve equivalent performance in terms of accuracy. Until epoch 25, it remains below 80%. We can explain this situation by referring to the observation made in **Figure 6**, which shows that the model continues to improve without falling into overfitting. This means that if we increase the number of epochs, the validation accuracy curve could reach performance levels close to those of the training phase, surpassing 90%.

Regarding **Figure 8**, the significant gain in execution time of our model can be attributed to its architecture. Indeed, we have designed an architecture with fewer layers (a total of 4 layers). Although the number of filters used in each layer may be high, having fewer layers [33], combined with the optimization techniques we applied (see **Table 1** and **Table 2**), accounts for this considerable time gain compared to ResNet and ShuffleNet. Moreover, with an architecture having fewer layers, it is logical that the memory required is reduced (see **Figure 9**). Although other factors influence memory consumption, the number of layers is one of the most determining factors.

Finally, although our model requires less execution time and relatively less memory space, it achieves high accuracy similar to ResNet, whose execution time is ten times longer than that of our model (see **Figure 10**). Additionally, its memory consumption is three times higher than that of our model (KKDNet). We attribute these results to the optimization techniques employed and the use of a cyclic learning rate in the architecture design.

5. Conclusions

In this research, we introduced a novel Convolution Neural Network which reduces computational time and memory consumption. We use it to classify brain tumor in four classes: glioma tumor, pituitary tumor, Meningioma tumor and no tumor.

The results obtained showed that our KKDnet model offers comparable performance to recent models in the literature, while requiring less computation time to run and less memory to store trainable and no-trainable parameters. It should be noted that we applied techniques such as the cyclic learning rate, which not only accelerated learning but also made the model more efficient. Ultimately, the results showed that our KKDNet model with fewer layers achieved superior performance in the identification and classification of brain tumor, and could therefore serve.

Since the training was conducted with a single dataset of 3264 images, it is possible that the model, despite its advantages, may experience a degradation in detection performance when used on new data. To address this situation, we plan to create a new dataset from MRI scans of hospitals in Ivory Coast while continuing to adjust the mode. Then we plan to develop a web application based on this model to assist professionals in developing countries in diagnosing brain tumors, addressing the shortage of specialists and the lack of well-trained professionals. In the future, to enhance the robustness of our model, we plan to expand the dataset and increase the number of epochs. We also intend to develop a cloud-based application to integrate the model.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

- [1] Yen, C., Lin, C. and Chiang, M. (2023) Exploring the Frontiers of Neuroimaging: A Review of Recent Advances in Understanding Brain Functioning and Disorders. *Life*, **13**, Article 1472. <https://doi.org/10.3390/life13071472>
- [2] Weiskopf, N., Edwards, L.J., Helms, G., Mohammadi, S. and Kirilina, E. (2021) Quantitative Magnetic Resonance Imaging of Brain Anatomy and *in vivo* Histology. *Nature Reviews Physics*, **3**, 570-588. <https://doi.org/10.1038/s42254-021-00326-1>
- [3] Kapoor, L. and Thakur, S. (2017) A Survey on Brain Tumor Detection Using Image Processing Techniques. 2017 *7th International Conference on Cloud Computing, Data Science & Engineering-Confluence*, Noida, 12-13 January 2017, 582-585. <https://doi.org/10.1109/confluence.2017.7943218>

- [4] National Cancer Institute (2024) Rare Brain and Spine Tumors. <https://www.cancer.gov/rare-brain-spine-tumor/tumors>
- [5] Schaff, L.R. and Mellinghoff, I.K. (2023) Glioblastoma and Other Primary Brain Malignancies in Adults. *JAMA*, **329**, 574-587. <https://doi.org/10.1001/jama.2023.0023>
- [6] Ostrom, Q.T., Gittleman, H., Xu, J., Kromer, C., Wolinsky, Y., Kruchko, C., *et al.* (2016) CBTRUS Statistical Report: Primary Brain and Other Central Nervous System Tumors Diagnosed in the United States in 2009-2013. *Neuro-Oncology*, **18**, v1-v75. <https://doi.org/10.1093/neuonc/now207>
- [7] Desai, M. and Shah, M. (2021) An Anatomization on Breast Cancer Detection and Diagnosis Employing Multi-Layer Perceptron Neural Network (MLP) and Convolutional Neural Network (CNN). *Clinical eHealth*, **4**, 1-11. <https://doi.org/10.1016/j.ceh.2020.11.002>
- [8] Iv, M., Yoon, B.C., Heit, J.J., Fischbein, N. and Wintermark, M. (2018) Current Clinical State of Advanced Magnetic Resonance Imaging for Brain Tumor Diagnosis and Follow Up. *Seminars in Roentgenology*, **53**, 45-61. <https://doi.org/10.1053/j.ro.2017.11.005>
- [9] Mamun, M., Bin Shawkat, S., Ahammed, M.S., Uddin, M.M., Mahmud, M.I. and Islam, A.M. (2022) Deep Learning Based Model for Alzheimer's Disease Detection Using Brain MRI Images. 2022 *IEEE 13th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, New York, 26-29 October 2022, 510-516. <https://doi.org/10.1109/uemcon54665.2022.9965730>
- [10] Yenet, A., Nibret, G. and Tegegne, B.A. (2023) Challenges to the Availability and Affordability of Essential Medicines in African Countries: A Scoping Review. *ClinicoEconomics and Outcomes Research*, **15**, 443-458. <https://doi.org/10.2147/ceor.s413546>
- [11] Abd-Ellah, M.K., Awad, A.I., Khalaf, A.A.M. and Hamed, H.F.A. (2019) A Review on Brain Tumor Diagnosis from MRI Images: Practical Implications, Key Achievements, and Lessons Learned. *Magnetic Resonance Imaging*, **61**, 300-318. <https://doi.org/10.1016/j.mri.2019.05.028>
- [12] Chatterjee, N., Duda, J., Gee, J., Elahi, A., Martin, K., Doan, V., *et al.* (2024) A Cloud-Based System for Automated AI Image Analysis and Reporting. *Journal of Imaging Informatics in Medicine*. <https://doi.org/10.1007/s10278-024-01200-z>
- [13] Latif, J., Xiao, C., Imran, A. and Tu, S. (2019) Medical Imaging Using Machine Learning and Deep Learning Algorithms: A Review. 2019 *2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, Sukkur, 30-31 January 2019, 1-5. <https://doi.org/10.1109/icomet.2019.8673502>
- [14] Jiang, S., Gu, Y. and Kumar, E. (2023) Magnetic Resonance Imaging (MRI) Brain Tumor Image Classification Based on Five Machine Learning Algorithms. *Cloud Computing and Data Science*, **4**, 122-133. <https://doi.org/10.37256/ccds.4220232740>
- [15] Ghosh, A. and Kole, A. (2023) A Comparative Study of Enhanced Machine Learning Algorithms for Brain Tumor Detection and Classification. Preprints. <https://doi.org/10.36227/techrxiv.16863136.v1>
- [16] Taha, A.M.H., Ariffin, S.B.B. and Abu-Naser, S.S. (2023) A Systematic Literature Review of Deep and Machine Learning Algorithms in Brain Tumor and Meta-Analysis. *Journal of Theoretical and Applied Information Technology*, **101**, 21-36. <https://www.jatit.org/volumes/Vol101No1/3Vol101No1.pdf>
- [17] Cong, S. and Zhou, Y. (2022) A Review of Convolutional Neural Network Architectures and Their Optimizations. *Artificial Intelligence Review*, **56**, 1905-1969. <https://doi.org/10.1007/s10462-022-10213-5>

- [18] Mahmud, M.I., Mamun, M. and Abdelgawad, A. (2023) A Deep Analysis of Brain Tumor Detection from MR Images Using Deep Learning Networks. *Algorithms*, **16**, Article 176. <https://doi.org/10.3390/a16040176>
- [19] Pedada, K.R., A., B.R., Patro, K.K., Allam, J.P., Jamjoom, M.M. and Samee, N.A. (2023) A Novel Approach for Brain Tumour Detection Using Deep Learning Based Technique. *Biomedical Signal Processing and Control*, **82**, Article 104549. <https://doi.org/10.1016/j.bspc.2022.104549>
- [20] Bhagyalaxmi, K., Dwarakanath, B. and Reddy, P.V.P. (2024) Deep Learning for Multi-Grade Brain Tumor Detection and Classification: A Prospective Survey. *Multimedia Tools and Applications*, **83**, 65889-65911. <https://doi.org/10.1007/s11042-024-18129-8>
- [21] Zhou, L., Wang, M. and Zhou, N. (2023) Distributed Federated Learning-Based Deep Learning Model for Privacy MRI Brain Tumor Detection. *Journal of Information, Technology and Policy*, **1**, 1-12. <https://doi.org/10.62836/jitp.2023.158>
- [22] Agarwal, M., Rani, G., Kumar, A., Kumar, K.P., Manikandan, R. and Gandomi, A.H. (2024) Deep Learning for Enhanced Brain Tumor Detection and Classification. *Results in Engineering*, **22**, Article 102117. <https://doi.org/10.1016/j.rineng.2024.102117>
- [23] Räuker, T., Ho, A., Casper, S. and Hadfield-Menell, D. (2023) Toward Transparent AI: A Survey on Interpreting the Inner Structures of Deep Neural Networks. 2023 *IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, Raleigh, 8-10 February 2023, 464-483. <https://doi.org/10.1109/satml54575.2023.00039>
- [24] Taye, M.M. (2023) Theoretical Understanding of Convolutional Neural Network: Concepts, Architectures, Applications, Future Directions. *Computation*, **11**, Article 52. <https://doi.org/10.3390/computation11030052>
- [25] Veeragandham, S. and Santhi, H. (2021) A Detailed Review on Challenges and Imperatives of Various CNN Algorithms in Weed Detection. 2021 *International Conference on Artificial Intelligence and Smart Systems (ICAIS)*, Coimbatore, 25-27 March 2021, 1068-1073. <https://doi.org/10.1109/icaiss50930.2021.9395986>
- [26] Shah, A., Shah, M., Pandya, A., Sushra, R., Sushra, R., Mehta, M., et al. (2023) A Comprehensive Study on Skin Cancer Detection Using Artificial Neural Network (ANN) and Convolutional Neural Network (CNN). *Clinical eHealth*, **6**, 76-84. <https://doi.org/10.1016/j.ceh.2023.08.002>
- [27] O'Shea, K. and Nash, R. (2015) An Introduction to Convolutional Neural Networks. arXiv: 1511.08458. <https://doi.org/10.48550/arXiv.1511.08458>
- [28] Bieder, F., Sandkühler, R. and Cattin, P.C. (2021) Comparison of Methods Generalizing Max- and Average-Pooling. arXiv: 2103.01746. <https://doi.org/10.48550/arXiv.2103.01746>
- [29] Jiang, Y., Xie, J. and Zhang, D. (2022) An Adaptive Offset Activation Function for CNN Image Classification Tasks. *Electronics*, **11**, Article 3799. <https://doi.org/10.3390/electronics11223799>
- [30] Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y. and Alsaadi, F.E. (2017) A Survey of Deep Neural Network Architectures and Their Applications. *Neurocomputing*, **234**, 11-26. <https://doi.org/10.1016/j.neucom.2016.12.038>
- [31] Yu, T. and Zhu, H. (2020) Hyper-Parameter Optimization: A Review of Algorithms and Applications. arXiv: 2003.05689. <https://doi.org/10.48550/arXiv.2003.05689>
- [32] Kaggle (2024) Brain Tumor Classification (MRI). <https://www.kaggle.com/datasets/sartajbhuvaji/brain-tumor-classification-mri>
- [33] Gao, Y., Liu, Y., Zhang, H., Li, Z., Zhu, Y., Lin, H., et al. (2020) Estimating GPU

Memory Consumption of Deep Learning Models. *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, Virtual, 8-13 November 2020, 1342-1352.
<https://doi.org/10.1145/3368089.3417050>