Brain tumor classification in magnetic resonance imaging images using convolutional neural network

Nihal Remzan, Karim Tahiry, Abdelmajid Farchi

Engineering, Industrial Management and Innovation Laboratory, Faculty of Sciences and Technics, Hassan First University, Settat, Morocco

Article Info	ABSTRACT
	Deep learning (DL) is a subfield of artificial intelligence (AI) used in several sectors, such as cybersecurity, finance, marketing, automated vehicles, and medicine. Due to the advancement of computer performance, DL has become very successful. In recent years, it has processed large amounts of data, and achieved good results, especially in image analysis such as segmentation and classification. Manual evaluation of tumors, based on medical images, requires expensive human labor and can easily lead to misdiagnosis of tumors. Researchers are interested in using DL algorithms for automatic tumor diagnosis. convolutional neural network (CNN) is one such algorithm. It is suitable for medical image classification tasks. In this paper, we will focus on the development of four sequential CNN models to classify brain tumors in magnetic resonance imaging (MRI) images. We followed two steps, the first being data preprocessing and the second being automatic classification of preprocessed images using CNN. The experiments were conducted on a dataset of 3,000 MRI images, divided into two classes: tumor and normal. We obtained a good accuracy of 98,27%, which outperforms other existing models.

Corresponding Author:

Nihal Remzan
Engineering, Industrial Management and Innovation (IMII) Laboratory, Faculty of Sciences and Technics, Hassan First University
Settat, Km 3, B.P.: 577 Route de Casablanca, Morocco
Email: n.remzan@uhp.ac.ma

1. INTRODUCTION

The brain is a complex organ composed of billions of body cells. Uncontrolled cell division results in the formation of an irregular group of cells called a "tumor" [1]. Primary and secondary brain tumors are the two types of tumors. Primary brain tumors can develop in the brain or in nearby tissues such as the nervous system and the membranes that cover the brain (meninges). Cancerous cells from other organs, such as the breast, lungs, kidneys, and so on, propagate to the brain, resulting in secondary brain tumors. Mutations in the DNA of primary brain tumors cause them to form. Normal cells die and cancerous cells grow as a result of these mutations. This damages the brain and is potentially fatal [2].

Medical image processing and analysis is extremely important in the domain of medicine. It has become one of the most useful techniques for detecting and identifying a variety of diseases. By visualizing and interpreting the image, it helps clinicians and radiologists to make a specific diagnosis [3]. Computed tomography (CT), positron emission tomography (PET) and magnetic resonance imaging (MRI) are types of medical imaging techniques. The level of advancement in such advanced imaging is so great that even tissue and microstructure may be visible during visual analysis [4]. MRI is a non-invasive and extremely reliable technique that allows visualization of the complex and interior regions of the human brain in great detail

without requiring human intervention. It is frequently used in the neurological system era to diagnose brain abnormalities such as tumors, cancers, lesions, and so on [3], [5].

In recent years, medical image classification has received a lot of attention. The term, "deep learning," is referring to the multi-layer artificial neural network (ANN). It has long been regarded as the most significant tool in recent decades, and has gained popularity due to its capacity to analyze huge amounts of data. Recently, the arrangement of deeper hidden layers has outperformed traditional methods in several areas [6], [7]. Convolutional neural network (CNN) is the best suitable deep learning model for medical image classification due to its many layers such as convolutional layer, pooling layer and fully connected layer [2], [8].

In recent years, a lot of study was conducted on brain tumor classification. The following is a short review of these works: Mohsen *et al.* [9] have exploited fuzzy C-means (FCM) segmentation, for separation of tumor and no-tumor regions of the brain. Furthermore, they extracted wavelet features using the discrete wavelet transform (DWT). Finally, they used a deep neural network (DNN) to classify brain tumors. This technique has shown high accuracy compared to traditional classification methods, such as linear discriminant analysis (LDA), sequential minimal optimization (SMO), and K-nearest neighbors (KNN). Islam *et al.* [10] have used a multi-fractal feature extraction (MultiFD) scheme to extract tissue texture features from brain tumors. An improved AdaBoost classification algorithm, has been developed to assess whether or not the brain tissue is tumoral. To classify each brain voxel into distinct groups, Huang *et al.* [11] have used a local independent projection classification (LIPC) system. The path function is also extracted using this tool. Cheng *et al.* [12] have developed a tumor classification method, that consists of two stages. The first one comprises the development of an offline database, in which tumor images are processed in sequential steps of tumor segmentation, features extraction, and distance metric learning. The second one involves online retrieval, or online learning. The input image is processed in the same way, and the extracted attribute is compared to distance metrics that have been studied and saved in an online database. He *et al.* [13] have classified the brain tumor utilizing DNN with auto-encoders. Before the image was processed by the DNN, it was subjected to segmentation and features extraction. Gray level co-occurrence matrix (GLCM), and DWT, were used to extract the image's intensity and texture-based features. And finally, for classification, DNN is used, which consists of two auto encoders and a softmax layer. Oza *et al.* [14] proposed a random forest classifier to classify benign and malignant brain tumors. Anitha and Murugavalli [15] developed a classification method for detecting tumor cells in the brain based on the KNN algorithm in their study. Thara and Jasmine [16] utilized fuzzy K-means clustering and C-means clustering to perform segmentation, then a neural network to perform image classification. Singh and Ansari [17] used the K-means algorithm and the normalized histogram segmentation technique to develop a brain tumor detection system. For the classification and accuracy of their system, they utilized the support vector machine (SVM) and the Naïve Bayes classifier. Halder and Dobe [18] proposed a brain tumor detection system for MRI image classification based on SVMs and rough K-means. Shil *et al.* [19] transformed MRI images into OtsoBinarization and then used K-means clustering segmentation to detect and classify brain tumors. They extracted the features using the discrete wavelet transform approach and utilized SVM for high-accuracy classification. To eliminate noise, Hamiane and Saeed [3] used DWT, followed by edge detection and segmentation. GLCM is utilized to extract the features, and the SVM is employed to classify tumors on MRI images. Angadi *et al.* [20] founded a special framework for identifying the cerebral tumor region from MRI data. The proposed framework uses a directed acyclic graph, Bayesian network, and neural network to optimize the performance of the input image after it has been subjected to a non-conventional segmentation technique.

This research presents a technique for classifying brain tumors based on MRI images. The proposed classification is based on a CNN. The objective of this research is to develop an automatic brain tumors classification system based on MRI images. This study differs from prior studies cited in [3], [9]–[20] due to the role of automatic features extraction performed by convolutional neural network, instead of manual features extraction by traditional algorithms. This study proved the success of utilizing CNN on medical image classification. In this work, we have developed four CNN models for brain tumor classification in MRI images. The rest of the article is organized: section 2 consists of a detailed description of the algorithm used and the dataset with its preprocessing. Section 3 presents the results and discussions. The conclusion of the article is presented in the last section.

2. METHOD
2.1. Convolutional neural network
CNN is a deep learning algorithm that attempts to emulate the brain's visual cortex when analyzing images. In the past, to achieve better classification results, most computer vision researchers extracted features manually. With the use of pooling layers and convolution layers, CNN now performs features

extraction automatically during the training phase [21]. CNN designates that the network uses convolution as a mathematical operation, instead of matrix multiplication as in ANN, in one or more of its layers [22]. CNN has three principal layers: the convolutional layer, followed by a non-linear activation function, the pooling layer, and the fully connected layer. In the following we will explain the concepts of each of these layers.

2.1.1. Convolutional layer

The convolutional layer is the heart of any convolutional neural network where the raw image is processed. A convolution is an operation with two functions. The first function presents the input values at a position in the image, and the second function presents a filter. The dot product between both functions gives the output. The filter then moves to the next position in the image, defined by the stride length. The computation repeats until the entire image is covered, producing a features map as shown in Figure 1. The filters are allowed to detect features in images [8], [23].
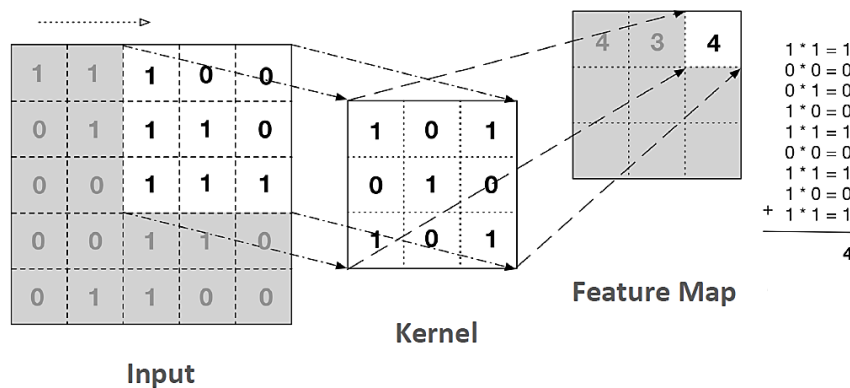

Figure 1. Example of a convolutional layer

For a CNN to learn more complex features, the convolutional layer must be deeper. For example, the first convolutional layer can learn to detect low-level features, such as the edges of an image. The second convolutional layer can learn to connect these edges to form circles or rectangles. Deeper convolutional layers can learn to detect more complex features such as the nose, and eyes [24].

The convolutional layer has several parameters to define itself. The first, Kernel-size, defines the size of the filter. For example, a filter of size 2×2 will have 4 weights. The second, Stride, specifies the number of pixels that should be moved in each direction, while applying convolution [24]. For the third, padding, must be "valid" or "same". For "valid", the convolutional layer does not apply zero padding, and some columns and rows can be ignored. For "same", the convolutional layer applies zero padding. In this case, the same number of output neurons as input neurons, divided by the stride, rounded up to the next integer higher. Zeros are then added around the input values [25].

At the convolutional layer's output, an activation function that is non-linear is used. For example, rectified linear unit (ReLU), is the most popular and widely used activation function in image classification tasks. This activation function zeroizes the negative input values, which speeds up and simplifies computation and learning as shown in Figure 2(a) [26], [27]. The (1) shows that:

$$f(x) = \max(0, x) = \begin{cases} x & if \quad x \geq 0, \\ 0 & if \quad x < 0. \end{cases} \tag{1}$$

In our experiments, we used another non-linear activation function for the output of convolutional layers, called scaled exponential linear units (SELU). This function is defined as a self-normalizing neural network. It produces normalized values, and is successfully applied to classification [27] as shown in Figure 2(b). The (2) shows that:

$$f(x) = \begin{cases} \lambda x & if \quad x > 0, \\ \lambda(\alpha \exp(x)) & if \quad x \leq 0. \end{cases} \tag{2}$$

where λ=1.0507 et α=1.6733 are the approximate values of SELU function parameters.

2.1.2. Pooling layer

The purpose of the pooling layer is to reduce the input image, in order to minimize the computation, the number of parameters, and memory usage. For this layer, the kernel size, stride and padding must be defined, as for the convolutional layer. However, a pooling kernel has no weight. It groups the input values using an aggregation function like max or average. Figure 3 shows a max pooling layer which is the most commonly used type [25].
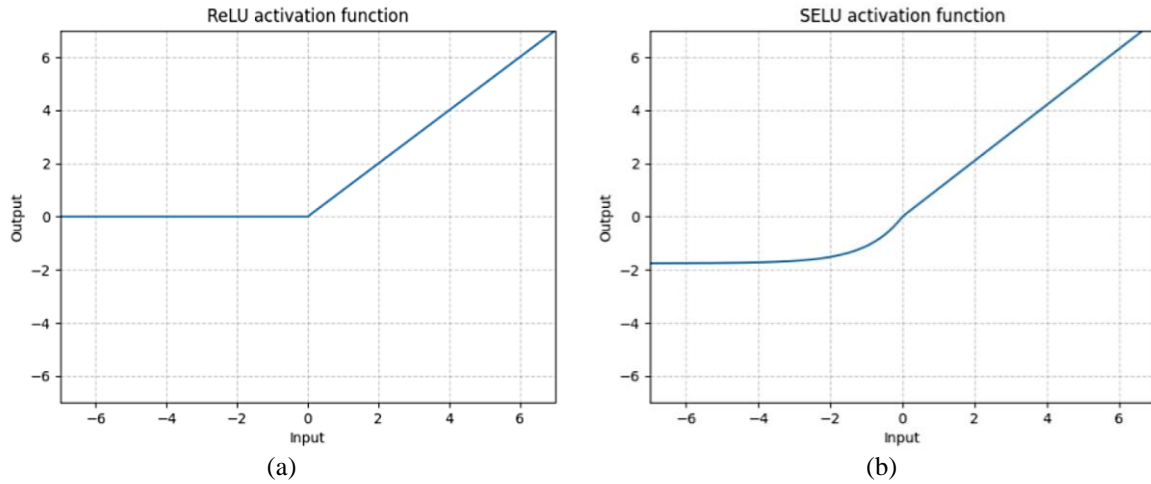

Figure 2. Description of two examples of activation functions (a) the ReLU activation function and (b) the SELU activation function
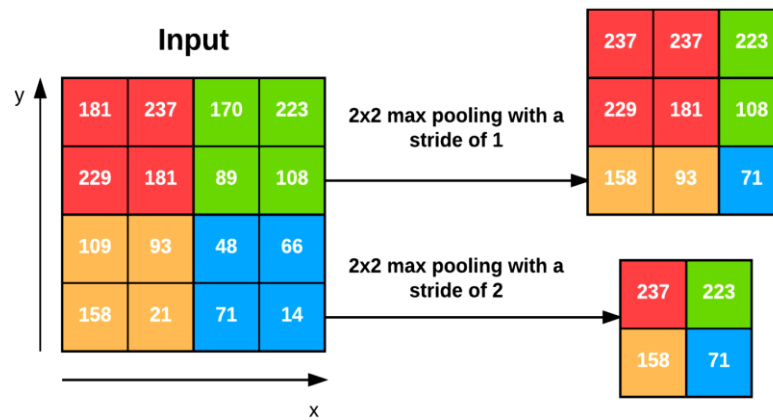
Figure 3. Max pooling layer with kernel size of 2×2, on top a stride of 1 and on bottom a stride of 2 [28]

2.1.3. Fully connected layers

Each neuron in the fully connected output layer will be connected to each neuron in the previous layer. The first layer of the fully connected layers is fed with the features extracted from the inputs. The output layer, generates a vector of dimension N, where N is the number of classes as shown in Figure 4. Each element of this vector presents the probability of belonging to a class. The softmax activation function presented in (3) is used to calculate these probabilities [26]–[28].

$$f(x_i) = \frac{\exp(x_i)}{\Sigma_j \exp(x_j)}$$ (3)

In this paper, we used four different CNN architectures, to test the accuracy of the classification of brain tumors. There are several pre-trained CNN architectures, such as AlexNet [29], VGG [30], and ResNet [13]. For this paper, the architectures are simpler (sequential) than those mentioned above.

2.2. Dataset

In our work, the dataset is shared by Hamada [31]. It includes 3,000 MRI images in different formats (jpg, JPG, png), and two modes, RGB (color image) and L (grayscale image). These images are divided into two classes (or categories), "yes" and "no". Table 1 shows examples for each class. The "yes" class includes 1,500 images of brains containing tumors, and the "no" class includes 1,500 images of healthy brains. Of the available images, we used only 2,891 images. 2,312 images (80%) in the training phase, and 579 images (20%) in the validation phase. To obtain these 2891 images, we followed several steps. The first step is to choose images in "jpg" format. The second step, is the resizing of the images in (128, 128), that is to say 128 in height and width. The third step, is to choose images in red, green, and blue (RGB) mode. The last step, is to assign to each resized image, the appropriate label in the form of a binary array, using the one-hot encoding technique [32].
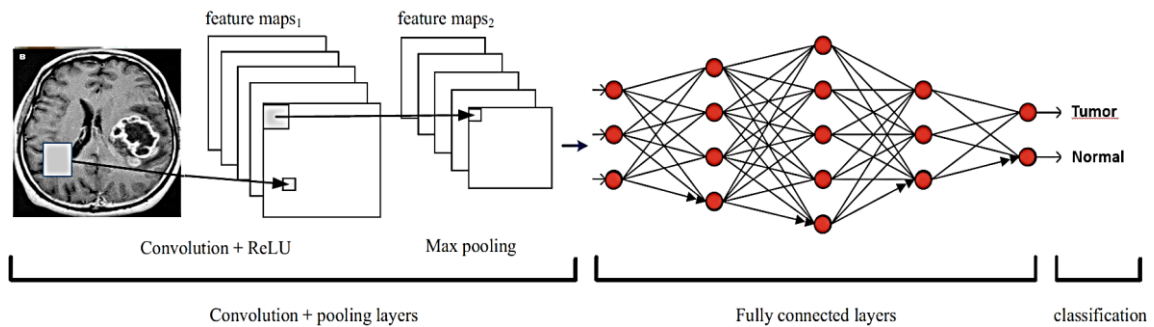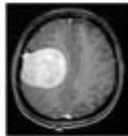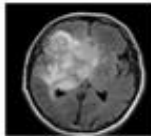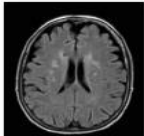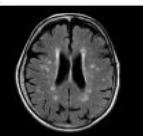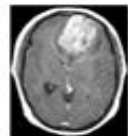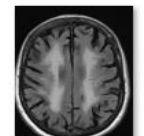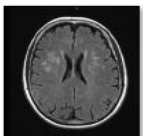

Figure 4. Simple example of a classification consisting of a convolutional layer + a ReLU activation function, a max pooling layer, and five fully connected layers

Table 1. System dataset

2.3. Performance evaluation metrics

The accuracy, precision, sensitivity and specificity of the algorithm's output were determined using true positive (TP), true negative (TN), false positive (FP), and false negative (FN) values as given in (4), (5), (6) and (7).

$$Accuracy = \frac{TP+TN}{TP+FN+FP+TN} \tag{4}$$

$$Precision = \frac{TP}{TP+FP} \tag{5}$$

$$Sensitivity = Recall = \frac{TP}{TP+FN} \tag{6}$$

$$Specificity = \frac{TN}{TN+FP} \tag{7}$$

True positives (TP) are cases where the predicted case was detected as a tumor and the actual case was also detected as a tumor; true negatives (TN) are cases when the predicted case was detected as normal and the actual case was also detected as normal; false positives (FP) are cases when the predicted case was detected as a tumor, but the actual case was detected as normal; false negatives (FN) are cases in which the predicted case is detected as normal while the actual case is detected as a tumor.

3.   RESULTS AND DISCUSSION

For our first experiment, we used two models that are different in depth. The first model is shown in Table 2, consists of two convolution layers, one max pooling layer, and two fully connected layers, also called dense layers. The second model, presented in Table 3, consists of four convolution layers, two max pooling layers and two dense layers. Both models receive matrices of the shape (128, 128, 32) as input. However, they share the same characteristics and layers: the number and size of filters in the convolutional layers are 32 and (2×2), respectively. The size of the kernel of the max pooling layers is 2×2, and the size of the output vector of the fully connected layers, is set to two (we have two classes). The softmax activation function is applied to the output, to convert a real vector into a categorical probability vector [27]. Other layers are used. Batchnormalization, takes the output of the precedent layer and normalizes it in batches. This layer helps to stabilize the training [28]. Dropout, is a regularization technique that can be applied to various layers' output such as the convolution, pooling, or fully connected layer. It randomly removes some neurons and their input and output connections during training to reduce overfitting [33]. The last characteristic is the Flatten layer, which converts the matrices into a vector. It is always placed just before the first dense layer, as the latter receives a vector as input.

Table 2. Model 1 parameters

Layers	Output Shape	Number of parameters
Conv_1	(None, 128, 128, 32)	416
Conv_2	(None, 128, 128, 32)	4128
Batchnormalization	(None, 128, 128, 32)	128
Maxpooling	(None, 64, 64, 32)	0
Dropout_1	(None, 64, 64, 32)	0
Flatten	(None, 131072)	0
Dense_1	(None, 512)	67109376
Dropout_2	(None, 512)	0
Dense_2	(None, 2)	1026

Table 3. Model 2 parameters

Layers	Output Shape	Number of parameters
Conv_1	(None, 128, 128, 32)	416
Conv_2	(None, 128, 128, 32)	4128
BatcHnormalization_1	(None, 128, 128, 32)	128
Maxpooling_1	(None, 64, 64, 32)	0
Dropout_1	(None, 64, 64, 32)	0
Conv_3	(None, 64, 64, 32)	4128
Conv2d_4	(None, 64, 64, 32)	4128
Batchnormalization_2	(None, 64, 64, 32)	128
Maxpooling_2	(None, 32, 32, 32)	0
Dropout_2	(None, 32, 32, 32)	0
Flatten_1	(None, 32768)	0
Dense_1	(None, 512)	16777728
Dropout_3	(None, 512)	0
Dense_2	(None, 2)	1026

3.1. Discussions and comparisons

The implementation of the two models generated four metric values as shown in Table 4. To evaluate the classification models' performance, accuracy, precision, sensitivity and specificity were calculated based on (4), (5), (6) and (7) using the values of TP, TN, FP, and FN. The most popular and widely used metric for evaluating the results is accuracy, which indicates the classification's overall accuracy. The models were trained without overfitting, as shown in Figures 5 and 6. Due to the techniques used to reduce overfitting, the values of loss (respectively accuracy) at training and validation decreased (respectively increased) together during training of both models without major differences between their calculated values during the twenty training epochs. Loss is calculated using the loss function [26], which measures the loss relative to the target. We used categorical cross-entropy as the loss function, since each

data point can only be assigned to one category [34]. From the comparative study of the two models presented in Table 4, we can see that model 2 generated the best results, compared to model 1. Therefore, we can deduce that CNN's depth is essential to obtain good results.

In order to have a better architecture, we try to vary the number of filters, in the convolution layers of model 2, into 64 filters. We present two architectures. In the first one, we vary the number of filters in all convolution layers into 64 filters (model 3). In the second, we vary the number of filters in the third and fourth convolution layers (model 4). Figures 7 and 8 show the loss and accuracy of model 3 and 4. Table 5, shows a comparison between the results obtained. We concluded that the number of filters influences the results of the classifier. However, increasing the number of filters does not always give the best performance of CNN as shown in model 3. Table 6 shows the final architecture.

Table 4. Comparison between the results of model 1 and model 2

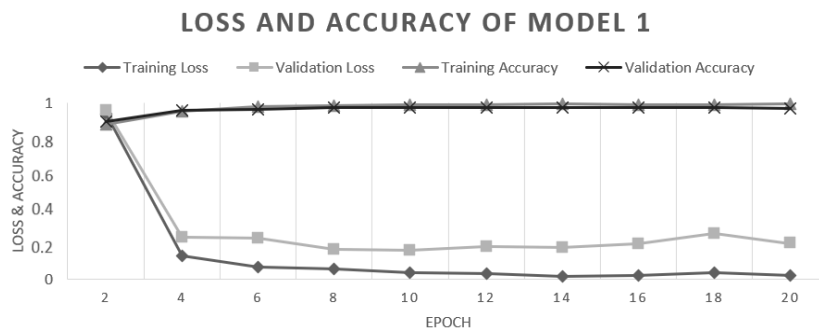| Model	Convolution	Max pooling	Dense	Accuracy	Loss	Precision	Sensitivity	Specificity
1	2	1	2	0.9654	0.2032	0.9640	0.9640	0.9667
2	4	2	2	0.9706	0.1528	0.9788	0.9746	0.9734

Figure 5. Loss and accuracy of model 1
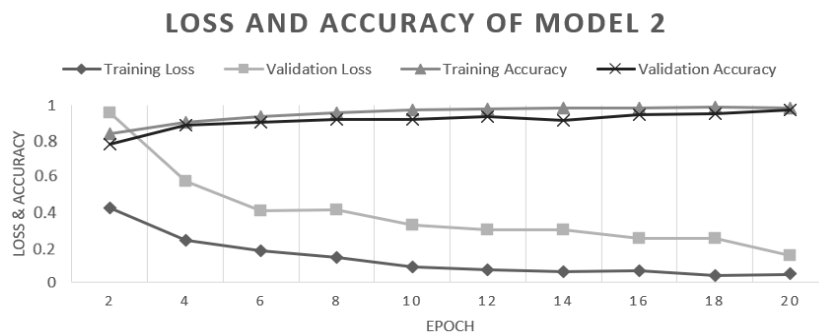

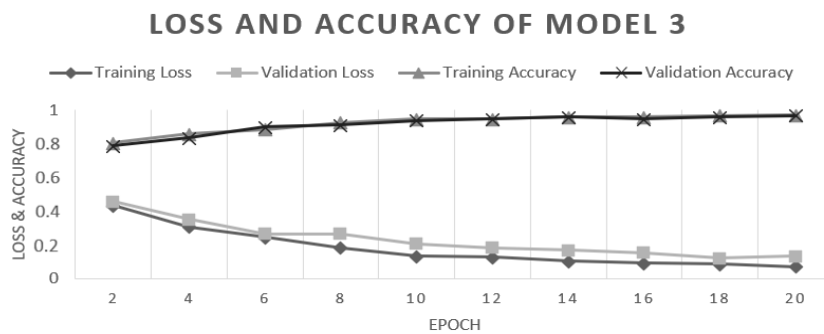Figure 6. Loss and accuracy of model 2

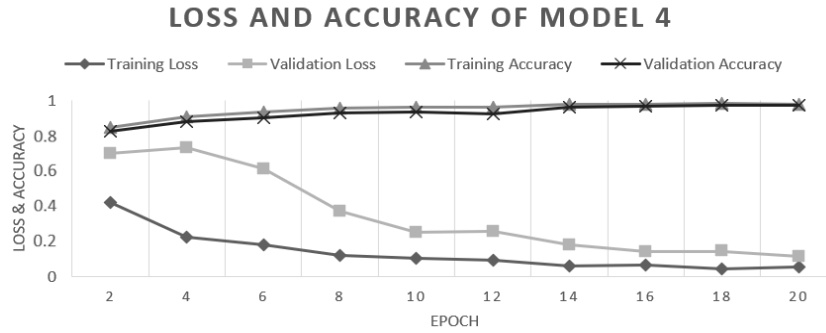Figure 7. Loss and accuracy of model 3

LOSS AND ACCURACY OF MODEL 4

Figure 8. Loss and accuracy of model 4

Table 5. Comparison between the results of model 2, model 3, and model 4

Model	conv_1	conv_2	conv_3	conv_4	Loss	Accuracy	Precision	Sensitivity	Specificity
2	32	32	32	32	0.1528	0.9706	0.9788	0.9746	0.9734
3	64	64	64	64	0.1359	0.9637	0.9604	0.9638	0.9635
4	32	32	64	64	0.1139	0.9741	0.9747	0.9712	0.9767

Table 6. Model 4 parameters

Layer	Output Shape	Number of parameters
Conv_1	(None, 128, 128, 32)	416
Conv_2	(None, 128, 128, 32)	4128
Batchnormalization_1	(None, 128, 128, 32)	128
Maxpooling_1	(None, 64, 64, 32)	0
Dropout_1	(None, 64, 64, 32)	0
Conv_3	(None, 64, 64, 64)	4128
Conv_4	(None, 64, 64, 64)	4128
Batchnormalization_2	(None, 64, 64, 64)	128
Maxpooling_2	(None, 32, 32, 64)	0
Dropout_2	(None, 32, 32, 64)	0
Flatten_1	(None, 65536)	0
Dense_1	(None, 512)	16777728
Dropout_3	(None, 512)	0
Dense_2	(None, 2)	1026

After choosing the best model (model 4), we tried to increase the number of epochs until we obtained the best possible results. Even if we were to increase the number of epochs, there would be no further improvement in the results. At each epoch, the model uses the optimizer (for our experiments, we used the Adamax optimizer [35]) to update its weights, which are randomly initialized at the beginning. This improves the final prediction, by minimizing the loss function [26]. In this experiment, we achieved the best performance, by choosing 30 epochs. Figures 9 and 10, show the loss and accuracy obtained for 25 and 30 epochs. Table 7 shows the results obtained from model 4 with 20, 25, and 30 epochs.
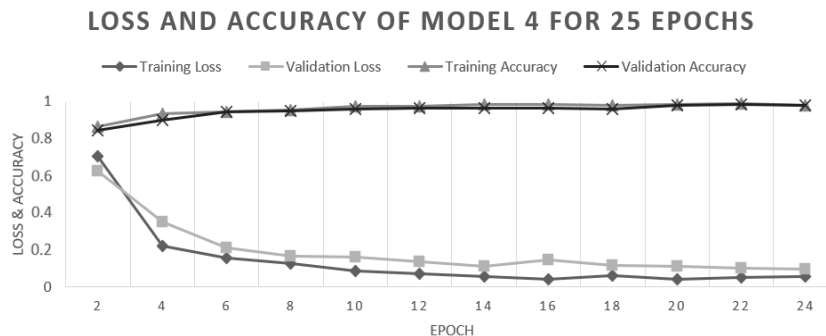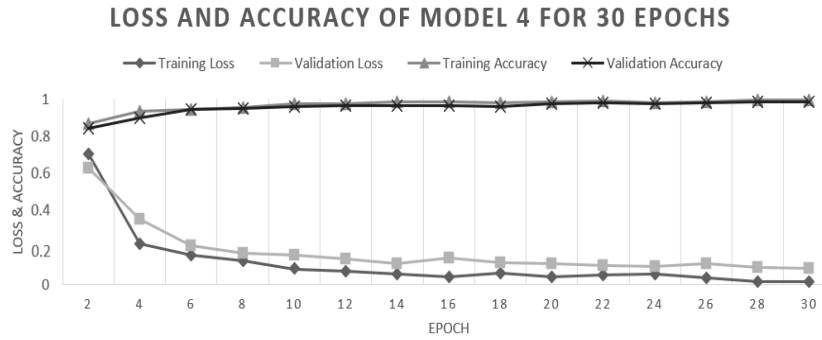
LOSS AND ACCURACY OF MODEL 4 FOR 25 EPOCHS

Figure 9. Loss and accuracy of model 4 for 25 epochs

LOSS AND ACCURACY OF MODEL 4 FOR 30 EPOCHS

Figure 10. Loss and accuracy of model 4 for 30 epochs

Table 7. Comparison of model 4 results with 20, 25, and 30 epochs

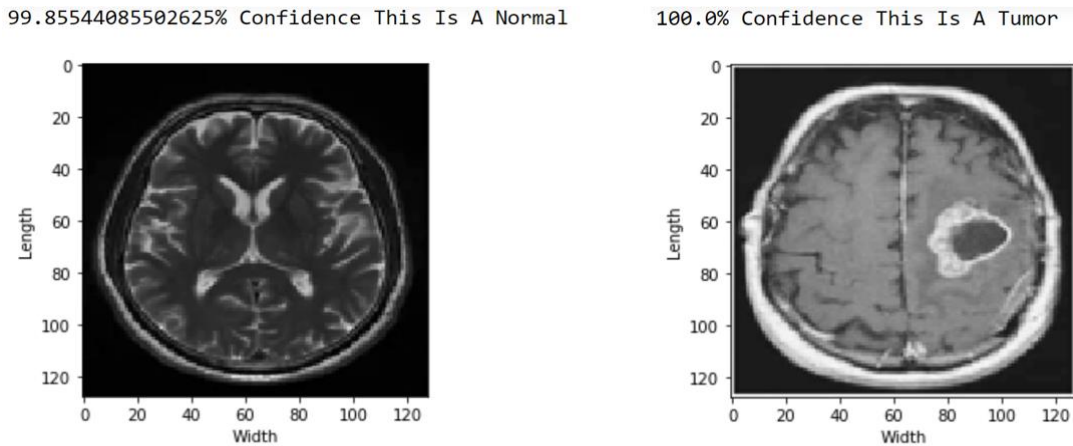| Epochs	Accuracy	Loss	Precision	Sensitivity	Specificity
20	0.9741	0.1139	0.9747	0.9712	0.9767
25	0.9758	0.1187	0.9748	0.9748	0.9767
30	0.9827	0.0857	0.9820	0.9820	0.9833

Figure 11. Test results on the new dataset

After training our final model, we used it to generate predictions on a new dataset never seen by our model. We loaded each of the images shown in Figure 11 and performed preprocessing for each image to fit our model. We then applied the prediction using our model. The test results are mentioned in Figure 11.

The accuracy of the proposed model is compared with other recently developed models for brain tumor classification. In these models, the authors used CNN [1], [2], [36]–[38], ANN [39], deep CNN (D-CNN) [40] and faster region-based CNN (R-CNN) [41]. Table 8 shows the accuracy obtained by each model. These accuracies range from 91.30% to 98.07%, all these values are lower than the accuracy obtained by our model.

Table 8. Comparison of the proposed model with others existing models

Model	Technique	Accuracy
[36]	CNN	91.30%
[41]	R-CNN	91.66%
[39]	ANN	92.14%
[2]	CNN	94.39%
[37]	CNN	96.08%
[38]	CNN	96.13%
[1]	CNN	97.50%
[40]	D-CNN	98.07%
Proposed model	CNN	98.27%

4.   CONCLUSION

In the medical sector, the classification of brain tumors is critical. We built four distinct CNN models in this research. We followed two general steps. The first is pre-processing of the raw data which consists of selecting the RGB images and resizing them, and the second is automatic feature extraction and classification by our CNN models. This paper showed that automatic feature extraction by a CNN performed well. In addition, changing parameters such as depth, number of filters in convolutional layers and number of epochs may significantly improve classification accuracy. The CNN models used have a large number of parameters, which can lead to overfitting. To solve this problem, regularization layers, called dropout, have been used. Our final model performed well, with an accuracy of 98.27% for the validation data set, outperforming other existing models.

REFERENCES

[1]   J. Seetha and S. S. Raja, "Brain tumor classification using convolutional neural networks," *Biomedical and Pharmacology Journal*, vol. 11, no. 3, pp. 1457–1461, Sep. 2018, doi: 10.13005/bpj/1511.

[2]   S. Das, O. F. M. R. R. Aranya, and N. N. Labiba, "Brain tumor classification using convolutional neural network," in *2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT)*, May 2019, pp. 1–5, doi: 10.1109/ICASERT.2019.8934603.

[3]   M. Hamiane and F. Saeed, "SVM classification of MRI brain images for computer-assisted diagnosis," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 7, no. 5, pp. 2555–2564, Oct. 2017, doi: 10.11591/ijece.v7i5.pp2555-2564.

[4]   T. S. Murthy and G. Sadashivappa, "Framework for comprehensive enhancement of brain tumor images with single-window operation," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 1, pp. 801–808, Feb. 2020, doi: 10.11591/ijece.v10i1.pp801-808.

[5]   S. Harish and G. F. A. Ahammed, "Integrated modelling approach for enhancing brain MRI with flexible pre-processing capability," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9, no. 4, pp. 2416–2424, Aug. 2019, doi: 10.11591/ijece.v9i4.pp2416-2424.

[6]   S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 International Conference on Engineering and Technology (ICET)*, Aug. 2017, pp. 1–6, doi: 10.1109/ICEngTechnol.2017.8308186.

[7]   T. Liu, S. Fang, Y. Zhao, P. Wang, and J. Zhang, "Implementation of training convolutional neural networks," *arXiv preprint arXiv:1506.01195*, Jun. 2015

[8]   J. Ker, L. Wang, J. Rao, and T. Lim, "Deep learning applications in medical image analysis," *IEEE Access*, vol. 6, pp. 9375–9389, 2018, doi: 10.1109/ACCESS.2017.2788044.

[9]   H. Mohsen, E.-S. A. El-Dahshan, E.-S. M. El-Horbaty, and A.-B. M. Salem, "Classification using deep learning neural networks for brain tumors," *Future Computing and Informatics Journal*, vol. 3, no. 1, pp. 68–71, Jun. 2018, doi: 10.1016/j.fcij.2017.12.001.

[10]   A. Islam, S. M. S. Reza, and K. M. Iftekharuddin, "Multifractal texture estimation for detection and segmentation of brain tumors," *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 11, pp. 3204–3215, Nov. 2013, doi: 10.1109/TBME.2013.2271383.

[11]   M. Huang, W. Yang, Y. Wu, J. Jiang, W. Chen, and Q. Feng, "Brain tumor segmentation based on local independent projection-based classification," *IEEE Transactions on Biomedical Engineering*, vol. 61, no. 10, pp. 2633–2645, Oct. 2014, doi: 10.1109/TBME.2014.2325410.

[12]   J. Cheng *et al.*, "Enhanced performance of brain tumor classification via tumor region augmentation and partition," *Plos One*, vol. 10, no. 10, Oct. 2015, doi: 10.1371/journal.pone.0140381.

[13]   K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[14]   M. Oza, R. Kapdi, and B. Student, "Brain tumor disease identification using random forest classifiers," *Brain*, vol. 7, no. 1, 2015

[15]   V. Anitha and S. Murugavalli, "Brain tumour classification using two-tier classifier with adaptive segmentation technique," *IET Computer Vision*, vol. 10, no. 1, pp. 9–17, Feb. 2016, doi: 10.1049/iet-cvi.2014.0193.

[16]   K. S. Thara and K. Jasmine, "Brain tumour detection in MRI images using PNN and GRNN," in *2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, Mar. 2016, pp. 1504–1510, doi: 10.1109/WiSPNET.2016.7566388.

[17]   G. Singh and M. A. Ansari, "Efficient detection of brain tumor from MRIs using K-means segmentation and normalized histogram," in *2016 1st India International Conference on Information Processing (IICIP)*, Aug. 2016, pp. 1–6, doi: 10.1109/IICIP.2016.7975365.

[18]   A. Halder and O. Dobe, "Rough K-means and support vector machine based brain tumor detection," in *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Sep. 2017, pp. 116–120, doi: 10.1109/ICACCI.2017.8125826.

[19]   S. K. Shil, F. P. Polly, M. A. Hossain, M. S. Ifthekhar, M. N. Uddin, and Y. M. Jang, "An improved brain tumor detection and classification mechanism," in *2017 International Conference on Information and Communication Technology Convergence (ICTC)*, Oct. 2017, pp. 54–57, doi: 10.1109/ICTC.2017.8190941.

[20]   P. Angadi, M. Nagendra, and H. M, "A novel framework for efficient identification of brain cancer region from brain MRI," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9, no. 2, pp. 1410–1417, Apr. 2019, doi: 10.11591/ijece.v9i2.pp1410-1417.

[21]   M. Y. Kamil, "A deep learning framework to detect Covid-19 disease via chest X-ray and CT scan images," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 11, no. 1, pp. 844–850, Feb. 2021, doi: 10.11591/ijece.v11i1.pp844-850.

[22]   I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. The MIT Press, 2016.

[23]   M. R. Al-Hadidi, B. AlSaaidah, and M. Al-Gawagzeh, "Glioblastomas brain tumour segmentation based on convolutional neural networks," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 5, pp. 4738–4744, Oct. 2020, doi: 10.11591/ijece.v10i5.pp4738-4744.

[24]   S. Pattanayak, *Pro deep learning with TensorFlow*. Berkeley, CA: Apress, 2017, doi: 10.1007/978-1-4842-3096-1.

[25]   A. Geron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, Inc, 2019.

[26]   J. Moolayil, *Learn keras for deep neural networks*. Berkeley, CA: Apress, 2019, doi: 10.1007/978-1-4842-4240-7.

[27] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation functions: comparison of trends in practice and research for deep learning," *arXiv preprint arXiv:1811.03378*, Nov. 2018

[28] A. Rosebrock, *Deep learning for computer vision with python-starter bundle*, 1st editio. Amazon, 2017.

[29] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.

[30] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *3rd International Conference on Learning Representations*, Sep. 2014

[31] A. Hamada, "Br35H :: brain tumor detection 2020." Kaggle, 2020. Accessed: Jul. 04, 2021. [Online]. Available: https://kaggle.com/ahmedhamada0/brain-tumor-detection

[32] C. Albon, *Machine learning with python cookbook: practical solutions from preprocessing to deep learning*. O'Reilly Media, Inc., 2018.

[33] I. Vasilev, D. Slater, G. Spacagna, P. Roelants, and V. Zocca, *Python deep learning: exploring deep learning techniques and neural network architectures with pytorch, keras, and TensorFlow*. Packt Publishing Ltd, 2019.

[34] Peltarion, "Categorical crossentropy loss function | Peltarion Platform," Peltarion.com. https://peltarion.com/knowledge-center/documentation/modeling-view/build-an-ai-model/loss-functions/categorical-crossentropy (accessed Apr. 25, 2021).

[35] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv:1609.04747v2*, Sep. 2016

[36] S. Sarkar, A. Kumar, S. Chakraborty, S. Aich, J.-S. Sim, and H.-C. Kim, "A CNN based approach for the detection of brain tumor using MRI scans," *Test Engineering and Management*, vol. 83, pp. 16580–16586, 2020

[37] C. L. Choudhury, C. Mahanty, R. Kumar, and B. K. Mishra, "Brain tumor detection and classification using convolutional neural network and deep neural network," in *2020 International Conference on Computer Science, Engineering and Applications (ICCSEA)*, Mar. 2020, pp. 1–4, doi: 10.1109/ICCSEA49143.2020.9132874.

[38] H. H. Sultan, N. M. Salem, and W. Al-Atabany, "Multi-classification of brain tumor images using deep neural network," *IEEE Access*, vol. 7, pp. 69215–69225, 2019, doi: 10.1109/ACCESS.2019.2919122.

[39] N. Arunkumar, M. A. Mohammed, S. A. Mostafa, D. A. Ibrahim, J. J. P. C. Rodrigues, and V. H. C. Albuquerque, "Fully automatic model-based segmentation and classification approach for MRI brain tumor using artificial neural networks," *Concurrency and Computation: Practice and Experience*, vol. 32, no. 1, Jan. 2020, doi: 10.1002/cpe.4962.

[40] M. Ganesan, N. Sivakumar, and M. Thirumaran, "Internet of medical things with cloud-based e-health services for brain tumour detection model using deep convolution neural network," *Electronic Government, an International Journal*, vol. 16, no. 1/2, 2020, doi: 10.1504/EG.2020.105240.

[41] E. Avşar and K. Salçin, "Detection and classification of brain tumours from MRI images using faster R-CNN," *Tehnički glasnik*, vol. 13, no. 4, pp. 337–342, Dec. 2019, doi: 10.31803/tg-20190712095507.

BIOGRAPHIES OF AUTHORS

Nihal Remzan was born in Casablanca, Morocco in 1996. She obtained the Bachelor of Sciences and Technics from the Faculty of Sciences and Technics, Hassan first University, Settat, Morocco in 2018 in Electronics, Electrotechnics, Automatic and the Master of Research from the National School of Applied Sciences, Abdelmalek Essaadi University, Tetouan, Morocco in 2020 in Signal Processing and Machine Learning. She is currently a Ph.D. student at Faculty of Sciences and Technics, Hassan first University since 2021 in Physics and Engineering Sciences in Engineering, Industrial Management and Innovation (IMII) Laboratory. She can be contacted at email: n.remzan@uhp.ac.ma.

Karim Tahiry was born in Settat, Morocco, in 1988. He received the Ph.D. degree from the Faculty of Sciences and Technics, University Hassan first, Settat, Morocco, in 2018 in Electronics and Telecommunications. He is currently Assistant Professor at Faculty of Sciences and Technics, Hassan first University since 2019. Member of Engineering, Industrial Management and Innovation (IMII) Laboratory. His research interests include speech recognition, signal processing, embedded systems, deep learning. He can be contacted at email: k.tahiry@uhp.ac.ma.

Abdelmajid Farchi Ing PhD in Electronics and Telecommunications Head of the "Signals and Systems" research team at the Laboratory of Engineering, Industrial Management and Innovation. Pedagogical manager of the Telecommunications and Embedded Systems engineering cycle at the Faculty of Science and Technology of Settat, Morocco. He can be contacted at email: abdelmajid.farchi1@gmail.com.