 ISSN 2579-7298

Brain Tumor Detection on Magnetic Resonance Imaging Using Deep Neural Network

Sutedi[a,1], Muh Royan Fauzi Maulana[b,1 *], Melda Agarina[b,2], Arman Suryadi Karim [b,3],Nurfiana [b,4]

[a] Institut Informatika dan Bisnis Darmajaya, Jl. ZA Pagar Alam, Bandar Lampung and 35214, Indonesia
[b] Institut Informatika dan Bisnis Darmajaya, Jl. ZA Pagar Alam, Bandar Lampung and 35214, Indonesia
[1] sutedi@darmajaya.ac.id*;[2] royanfauzimaulana25@gmail.com;[3] agharina@darmajaya.ac.id,[4] armansuryadi@gmail.com, ,[5] nurfiana@darmajaya.ac.id
* corresponding author

ARTICLE INFO

Article history
Received 29 Jul 2023
Revised 08 Sep 2023
Accepted 12 Des 2023

Keywords:

Machine learning
Deep learning
Brain tumor detection
CNN

ABSTRACT

Cancer is a heterogeneous disease that can attack all parts of the body. Cancer is caused by the abnormal and uncontrolled growth of body cells, resulting in damage to body tissue and the potential to cause death. Cancer is a type of malignant tumor that attacks the body, one of which is the brain. Every year there are 300 brain tumor patients in Indonesia, both adults and children. Generally, doctors use two methods to diagnose these tumors, namely: biopsy and magnetic resonance imaging (MRI). Although the use of biopsy is quite accurate in identifying brain tumors, the time required is relatively long, reaching 15 days. While using MRI is relatively fast, the resulting accuracy is low and depends on the experience of medical personnel. This research aims to develop a method for diagnosing brain tumors using MRI images to make it faster and more accurate. The method used in this research was a deep neural network with a convolutional neural network (CNN) architecture layer. This method was chosen because deep learning has the advantage of pattern recognition with a high level of accuracy and is directly proportional to the number of datasets. This study used a dataset of 300 MRI images with two-dimensional (2D) axial imaging. The metrics used as a basis for the performance of the deep neural network model are accuracy, sensitivity, specificity, precision, and dice similarity coefficient with the results of each metric, namely: 99.3%, 98.6%, 98%, 98%, 98.3%. The research results showed that a deep neural network can speed up the diagnosis of brain tumors with high accuracy within 0.2 seconds.

1. Introduction

Cancer is a heterogeneous group of diseases that can affect all parts of the body and has many anatomical and molecular sub-types. Each part of the body affected by cancer requires special treatment and diagnosis methods [1]. Cancer in general is a disease caused by abnormal and uncontrolled growth of body cells that results in damage to body tissues and potential death. Every year there are 14 million patients diagnosed with cancer worldwide, making cancer the second leading cause of death in the world [1].

Cancer is a type of malignant tumor [2]. One of the malignant tumors that is often found is a brain tumor that attacks the cell tissue of the brain membrane. Brain tumor disease is an abnormal growth of brain cells in or around the brain unnaturally and uncontrollably [3]. Generally, brain tumors are divided into two, namely, primary brain tumors and secondary brain tumors [3]. Primary brain tumors are abnormal and uncontrolled cell changes that originate from the brain cells themselves. Meanwhile, secondary brain tumors are abnormal cell changes resulting from cancer cells from other parts of the body [3].

Brain tumor cases in the world are increasing every year. About 300 patients in Indonesia are diagnosed with brain tumors each year [3]. Not only adults, but brain tumors also attack relatively young children [3].

To date, various ways have been developed to diagnose tumors. The anatomical approach of health images using Computerized Tomography Scan (CT Scan) and Magnetic Resonance Imaging (MRI) is a method that is currently commonly used to detect the presence of tumors, especially brain tumors. The advantage of using MRI (magnetic resonance images) over CT-Scan is that the image information provided in the image is more in-depth and clearly differentiated between soft tissue and hard tissue (bone) contained in the brain [4]. Thus, the results of an MRI can provide a clearer picture for doctors to evaluate and diagnose. In addition to MRI, another method with a higher level of diagnostic accuracy is the biopsy method [3]. However, this method takes 10-15 days to obtain results from the laboratory. MRI offers a relatively faster diagnosis time compared to the biopsy method, but its accuracy still needs to be improved. Brain tumor treatment requires a high level of accuracy and speed. Therefore, it is necessary to develop alternative methods that can reduce the error rate and can help doctors make quick and careful decisions [4].

There has been a lot of research and utilization of artificial intelligence (AI) technology in the field of medicine. This technology can be a potential alternative solution to solve the problem of brain tumor diagnosis. Machine learning and deep learning approaches that utilize deep neural network architecture have a good ability to detect patterns in images. Machine learning methods with deep wavlet architecture have been used to classify brain tumors in research by Isselmou Abd El Kader, Guizhi Xu, Zhang Shuai, Sani Saminu, Imran Javaid, Isah Salim Ahmad, and Souha Kamhi with an accuracy rate of 99.3% [5].

Based on this, this research takes the initiative to utilize computer vision technology in recognizing patterns on MRI to produce prototype tools that can increase the speed and accuracy of the brain tumor diagnosis process. This research is titled "brain tumor detection on magnetic resonance images with deep neural network method". It is hoped that the results of this research can minimize the level of misdiagnosis and doctors can focus more on the prognosis process for the patient's recovery.

Research referring to [3] by Andre R, Wahyu P, and Purbaningtyas R in 2021 with the title "Classification of brain tumors using convolutional neural networks with efficientnet-B3 architecture" examines the classification of brain tumors of the meningioma type and glioma. The dataset consists of 4091 images consisting of 394 images of glioma tumors, 827 images of meningioma, and 2870 images of pituitary tumors which are classified using the convolutional neural network method with the pre-trained efficientnet-B3 model which is included in the deep learning category. The results of testing the model with a confusion matrix, the highest accuracy reached 99.7% and the F1-Score value reached 99.7%. The efficientnet-B3 architecture applied in this research has the advantage of achieving high accuracy and can reduce the parameters of FLOPS (Floating Point Operations Per Second) which improves the ability of a model.

Research with the title "Classification of image patterns in brain tumor patients based on artificial neural networks (Case study of curative handling of brain tumor patients)" by Sri Hewanurweni, Budiani Destyningtias, and Andi Kurniawan Nugroho in 2018 applied the backpropagation method to classify brain tumors based on MRI images. Model testing results using MSE (Mean Squared Error) resulted in 96.55% accuracy [4]. However, the model that was trained only used 29 brain images which made the model not well generalized.

Research conducted by Ahmad Saleh, Rozana Sukaik, and Samy S with the title "Brain tumor classification using deep learning" applies deep learning methods with pre-trained xception models as its architecture in the case of brain tumor classification of pituitary, glioma and meningioma. The dataset used is 4480 images consisting of 2760 brain images diagnosed with tumors and 1720 images in a healthy state. Test results on the model using the F1-Score metric achieved an accuracy value of 98.75% [6].

Research referring to [5] by Isselmou Abd El Kader, Guizhi Xu, Zhang Shuai, Sani Saminu, Imran Javid, Isah Salim Ahmad with the title "Brain tumor detection and classification on MR images by a deep wavelet auto-encoder model" applies three stages that need to be passed by the image, namely: high pass filter, median filter and DWAE model on magnetic resonance image (MRI) based brain

tumor detection. This study uses five types of brain tumor MRI image datasets consisting of: BRATS 2012, BRATS 2013 BRATS 2014, 2015 challenge and BRATS 2015 and ISLES. The results of model testing using the confusion matrix evaluation metric with an average accuracy value of 99.3%. The method produced in this study can analyze a large number of images very well so that the model can predict new data.

2. Method

2.1. Research Flow

In general, this research is conducted to develop a model that can be developed to accelerate and improve accuracy in detecting brain tumor diseases based on MRI images. The stages of research conducted to develop the required machine learning model according to Aurélien Géron [7] are as follows.

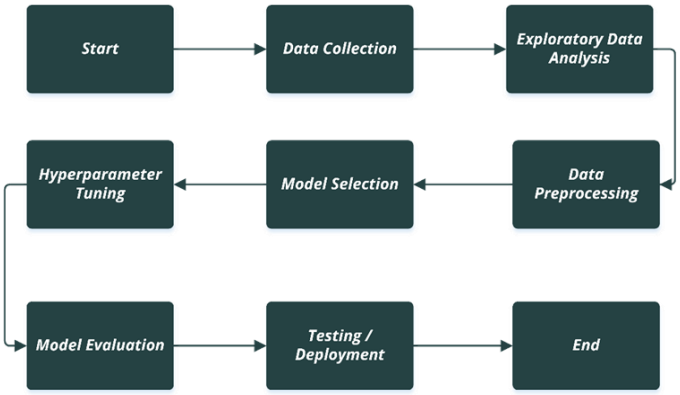

Fig.1. Research flow

2.2. Data Collection

This study uses MRI image datasets obtained from the Kaggle repository. The dataset used is 3000 images consisting of 1500 images diagnosed with meningioma brain tumors and 1500 images in a healthy state. The MRI images used in the dataset have an average dimension of 300 x 300 pixels with RGB image type.

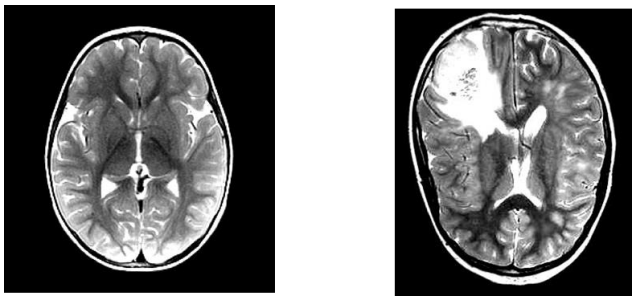

Fig. 2.. MRI brain scan

2.3. Exploratory Data Analysis

The collected data needs to be analyzed first to determine the characteristics of the dataset such as image distribution. MRI scans of brain tumors and healthy images will be randomly displayed to determine the general shape of the tumor.

2.4. Data Preprocessing

Before training, the dataset will be preprocessed into a specific format for better model performance. The dataset will be divided into training, validation, and testing with a ratio of 70:20:10. Then, image data that has pixel values in the range between 0 - 255 will be normalized

into the 0 - 1 value range to simplify the learning process and make it easier for the model to reach convergence [8].

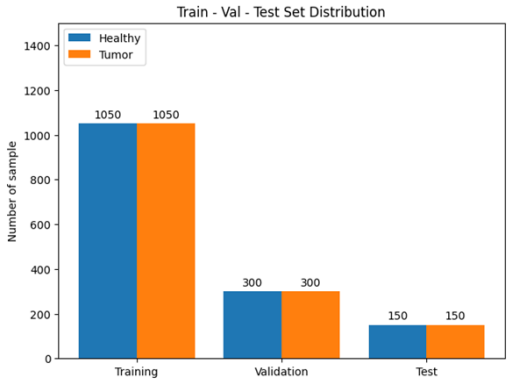

Fig. 3. Train-val-test set distribution

To change the value of each pixel in the image, this research uses the image data generator library from the tensorflow framework to normalize the pixel value into the range 0 - 1 and resize the image resolution to 150 x 150 so that the model is trained with a consistent image size.

The batch size is set to 16 for training data, validation data and testing data. Batch size functions as a hyperparameter that regulates the amount of data entered at once into the model to be trained. While class mode is set to a binary value because of the classification of two classes.

2.5. Model Selection

Deep learning is a branch of machine learning that applies renewable learning patterns in learning a representation of data. Deep learning emphasizes layer-by-layer learning that increases the representation of data to be more meaningful at each layer. These layers form a learning model called a deep neural network. Deep neural networks have more than two layers of neurons called hidden layers that receive weights from each layer of nodes or neurons [8].

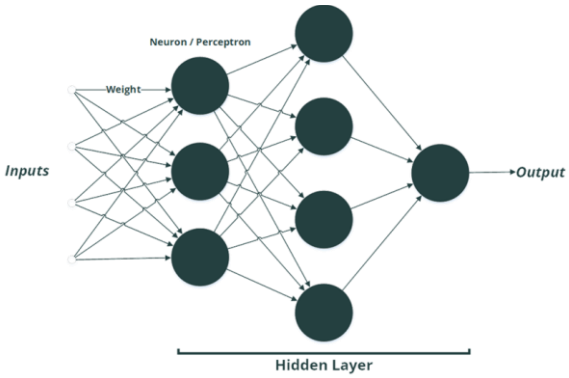

Fig. 4. Artificial neural network layers

In the hidden layer, there is a convolution layer that functions as a feature extraction. The layer that functions as a feature extraction is an integral part of the deep learning model that makes the deep neural network model able to find the most important features of data automatically and then train it [9]. One of the deep neural network models is convolutional neural network.

Convolutional neural network is one of the deep neural network models that has many layers including convolution layers, pooling layers, non-linearity, and fully connected layers. CNN has good performance in handling image data [10].

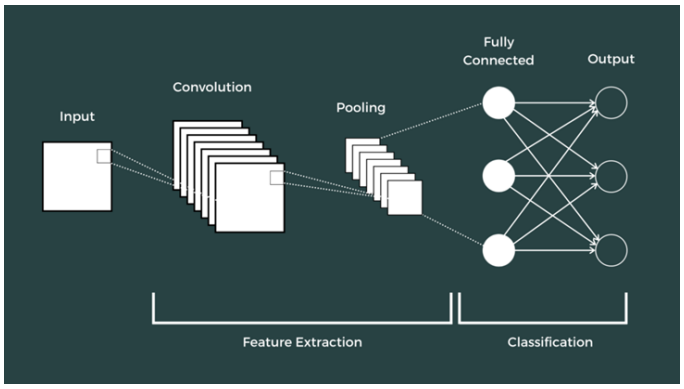

Fig. 5. Convolutional neural network

Good CNN performance is influenced by the convolution layer that will extract pixels in the image by multiplying each m x j pixel with a filter or kernel with a shift of each i x j pixel in the entire image matrix as many as n. The convolution result will be forwarded to the pooling layer to reduce the size of each image with the maxpool function that takes the highest pixel value from each i x j pixel. The convolution results will be forwarded to the pooling layer to reduce the pixel size of each image with the maxpool function that takes the highest pixel value from each i x j pixel. Mathematically, the convolution process can be denoted by formula (1) and the maxpooling process by formula (2).

$$z^l = h^{i-l} * W^l \quad \text{.....(1)}$$

With z^l is the output of the next layer, h is the input image, and W the kernel or filter matrix.

$$h_{xy}^l = max_{i=0,j=0}(0,x) \quad \text{.....(2)}$$

With $h_{xy}^l$is the *output*, x the input pixel value.

Images that have passed the convolution and pooling process will have a smaller image size than the original size but are still rich in features. Then the process continues to the fully connected layer where each input will be calculated the weight on each neuron. In the fully connected layer, the weight value is calculated using the backpropagation algorithm which mathematically each neuron is denoted by formula (3). Each layer in the fully connected layer has an activation function that functions to capture complex data values and pass them on to the next neuron. The activation function used by each neuron is Relu (Rectified Linear Unit) (4) and the output layer uses a sigmoid activation function (5) to predict two classes, namely the class of images diagnosed with brain tumors and healthy brain images [11].

$$a_j^{(i)} = f(\sum_{i=0}^{n} \quad w^{(i)} a_j^{(i-l)} + b_j^{(i)} \quad \text{.....(3)}$$

Description :

$a_j^{(i)}$: Output weight of the neuron

f : Activation function

w : Weight

a : Input value of the previous neuron

b : Bias neuron

$$ReLU(z_i) = max(0,z) \quad \text{.....(4)}$$

Description :

z_i : Backpropagation calculation result

$$f(z) = \sigma(z) = \frac{1}{1+e^{-z}} \text{.....(5)}$$

Description :

z : Output probability

To achieve high accuracy with low loss, the architecture of the deep neural network model needs to have ideal layers. The model is said to be ideal or (good fit) when it is between overfitting and underfitting, and between undercapacity and overcapacity [8]. The architecture of the deep neural network model proposed in this study is a convolutional neural network with details of each layer consisting of several convolutional blocks as presented in table 1.

Table 1. Block Convolutional Architecture

Block Convolutional	
Layer	**Layer Type**
1	Convolutional 2D Layer
2	Batch Normalization Layer
3	Activation: ReLu Layer
4	Convolutional 2D Layer
5	Batch Normalization Layer
6	Activation: ReLu Layer
7	Max Pooling Layer

Table 2. Proposed architecture

Block Convolutional	
Layer	**Layer Type**
1-7	Convolutional Block 1: (128), Kernel Size: (3 x 3), Input shape: 150 x 150 x 3
8-14	Convolutional Block 2: (64), Kernel Size: (3 x 3),
15-21	Convolutional Block 3: (32), Kernel Size: (3 x 3),
22-28	Convolutional Block 4: (16), Kernel Size: (3 x 3),
29-35	Convolutional Block 5: (8), Kernel Size: (3 x 3),
36	Gloval Average Pooling
37	Dense Layer, Neruron: 128, Activation Function: ReLU
38	Dropout Layer (0.2)
39 (Output Layer)	Dense Layer, Neruron: 1, Activation Function : Sigmoid

Next, the model that has been designed will be trained. In the process of training the model, the loss function and optimizer need to be determined first to regulate how the backpropagation algorithm works. The loss function acts as a function that returns the value of the distance or gap between the predicted results and the actual results to improve the weight of each neuron so that the model can recognize patterns in the image [8]. In this study, the case studied has two classes, namely MRI images diagnosed with tumors and healthy. Thus, the loss function used is binary crossentrophy (6). Furthermore, the loss value

will be passed into the optimizer function as a predictor. The optimizer has a role to predict and learn image patterns with a predetermined learning rate and loss value that improves how the optimizer predicts images [8]. In this research, the optimizer used is Adam (7) which provides output in the form of probability prediction results.

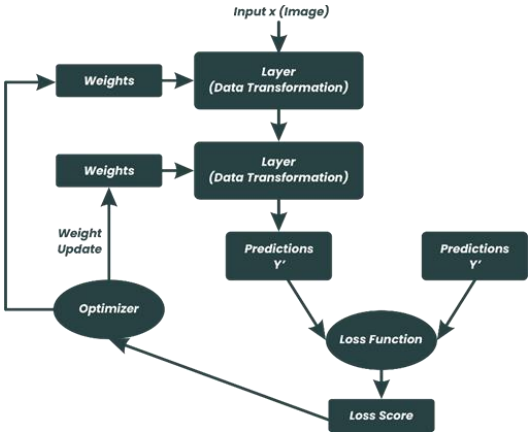

Fig. 6. Relationship between weights, layers, optimizer and loss

$$-(y\ log(p)\ +\ (1-y)log(1-p)) \tag{6}$$

Description:

y : binary indicator (0 or 1)

p : prediction probability

$$W_{t-1} = W_t - \alpha\, m_t \tag{7}$$

where,

$$m_t = \beta m_{t-1} + (1-\beta)[\frac{\partial L}{\partial W_t}] \tag{8}$$

Description:

m_t = aggregation of gradients at time t [running] (initialization, mt = 0)

m_{t-1} = aggregated gradient at time t-1 [previous]

W_t = time weight t

W_{t-1} = weight of time t+1

αt = learning rate at run time t

∂L = loss function

∂Wt = weight sharing at time t

β = average moving parameter (const, 0.9)

The training process will be done with a hard library by defining the loss and optimizer. Then, the model will be trained for 50 epochs with validation data. The epoch hyperparameter plays a role in determining how many iterations are performed from the first layer to the end and then back to the first layer in the model training process.

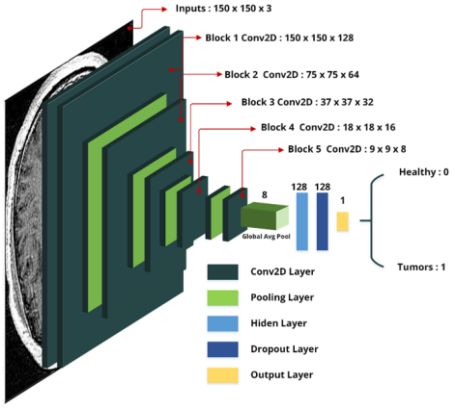

Fig. 7. Model Architecture

2.6. Hyperparameter Tuning

The hyperparameter tuning or hyperparameter optimization process plays a role in determining the optimal set of hyperparameter values for the model. These include the number of epochs, the number of neurons in the dense layer, the number of convolutional filters, and the appropriate learning rate for the model. In determining the most optimal hyperparameters, the tuning algorithm will conduct a series of trials that combine various values of each neuron, learning rate, epoch, and convolutional filter, which then the model will be referred to as a hypermodel [8].

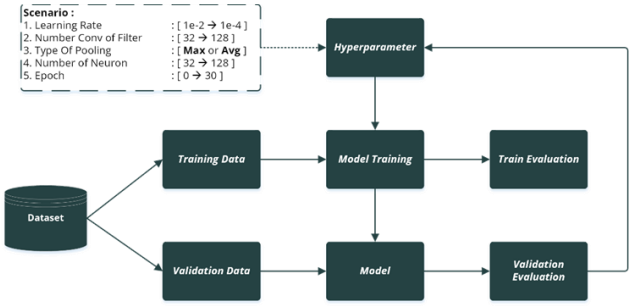

Fig 8. Hyperparameter tuning mechanism

Table 3. set of hyperparameter values tested

Hyperparameters	Value
Learning rate	[1e-2, 1e-3, 1e-4]
Epoch	Maximum 30 epochs for each test
Neuron	minimum 32 and maximum 128 with an increment of 32 neurons per trial.
Number of convolution filters	32 - 128

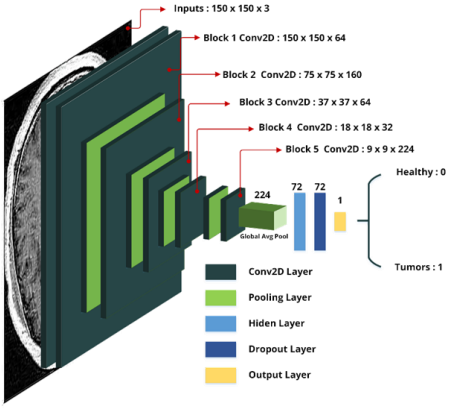

Fig. 8. Hypermodel architecture

2.7. Model Evaluation

The evaluation metrics used to evaluate the accuracy of the model are accuracy, sensitivity, specificity, precision and F1-Score and Dice similarity Coefficient. Model evaluation metrics are obtained from calculating the value of True Positives (TP), True Negatives (TN), False Positives (FP) and False Negatives (FN). The trained model will predict the testing dataset to get the final metric value from the total images that have been successfully predicted.

Accuracy is the ratio of the number of correctly classified samples to the total number of samples [12].

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{9}$$

Sensitivity/recall /True Positive Rate (TPR) is the ratio of samples that are classified as true positive out of all samples diagnosed with the disease [13].

$$Sensitivity = \frac{TP}{TP+ FN} \tag{10}$$

Specificity is the ratio of samples that are classified negatively correctly from all samples that are not diagnosed with the disease [12].

$$specifity = \frac{TN}{TN+ FP} \tag{11}$$

Precision / Positive Predicted Value (PPV) is the proportion of the number of samples classified as true positive out of all samples classified as positive [12].

$$precision = \frac{TP}{TP+ FP} \tag{12}$$

F1-Score is the harmonic mean of precision and sensitivity (recall) using statistical indices or confusion matrix [6].

$$F1\ Score = \frac{precision\ x\ recall}{precision + recall} = \frac{2\ TP}{2\ TP + FP + FN} \tag{13}$$

Dice similarity coefficient is the ratio between the actual value of tumor and non-tumor compared to the predicted value of tumor and non-tumor [5].

$$DSC = \frac{2\ TP}{FP + 2TP + FN} \tag{14}$$

3. Results and Discussion

Based on the methodology that has been designed in the case of brain tumor detection in MRI images with the Deep Neural Network method, the following results are obtained.

The data that has been collected through the Kaggle repository platform totals 3000 brain images which are the results of Magnetic Resonance Imaging scans with axial orientation. the size of the images in the dataset is relatively diverse with a size range of 200 x 250 pixels to 500 x 400 pixels.

Tumor-diagnosed images are seen to have a pattern resembling a circle within the brain area that characterizes brain abnormalities. The shape of the abnormality will be the main feature in predicting the presence of tumors in the human brain extracted through convolutional layers.

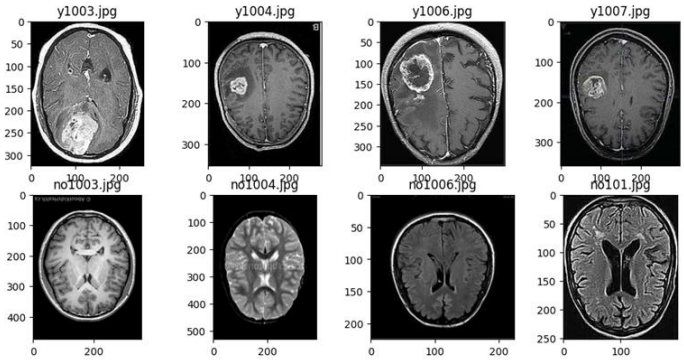

Fig. 9. Sample Brain Dataset

Based on Figure 9 above, it can be seen that the image size in the dataset is relatively diverse with a size range of 200 x 250 pixels to 500 x 400 pixels.

Tumor-diagnosed images are seen to have a pattern resembling a circle within the brain area that characterizes brain abnormalities. The shape of the abnormality will be the main feature in predicting the presence of tumors in the human brain extracted through convolutional layers.

Based on the exploration that has been done, the data will be divided into three subsets, namely the train set, validation set, and testing set. The subset division is intended to train the model and test the model. The ratio between each subset train : validation: testing is 70: 20: 10. So out of 3000 images, 2100 images are allocated for the training set, 600 images for the validation set, and 300 images for the testing set, respectively.

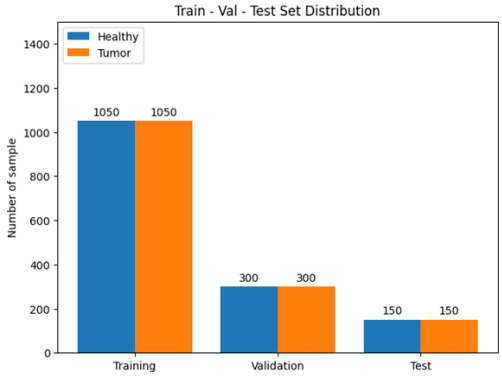

Fig. 9. Train- Val - Test distribution

Data that has been divided into three sets will be normalized to make it easier for the model to reach convergence. Normalization is performed on each image by dividing the value of each pixel by the maximum pixel value of 255 which makes the value of each pixel in the value range 0 - 1.

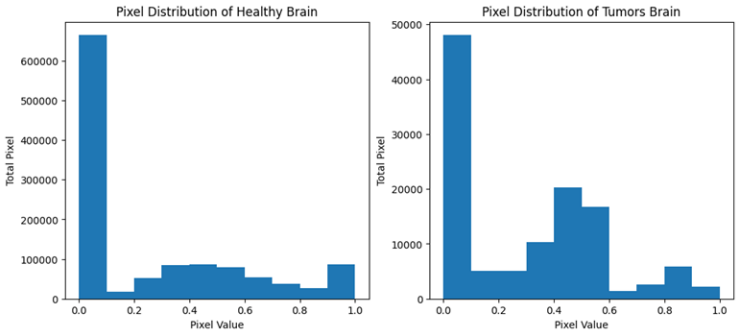

Fig. 10. Pixel distribution of MRI dataset

The model used in the research is Deep Neural Network which consists of Convolutional layers, Max Pooling layers and Dense (Neuron) layers. These layers have hyperparameters that function as operating values in each layer.

The convolution layer operates by multiplying each convolution filter with the input image to produce n images, where n is the number of filter values initialized in the hyperparameters.

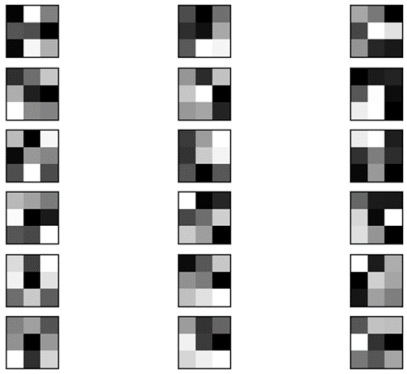

Fig. 11. Convolutional Filter

Each image resulting from the multiplication of these filters will provide a new image with different characteristics, such as an image that only shows the edges of the image, the background of the image, or the object in the image.

The first convolution block layer will receive direct input from the input image with a shape of 150 x 150 x 3, where at the end of the first convolution block layer will reduce the size of the image that has captured image features and provide an output of 75 x 75 x 128 images.

In the second layer, the convolutional block will receive the output from the previous layer to get the image feature value again and reduce it to reduce the image size and leave the most important part of the image.

The second convolutional block receives 75 x 75 x 128 inputs and outputs 64 new images with 37 x 37 x 64 outputs that have been multiplied by the convolutional filter.

The third convolutional block again takes the output image from the previous layer as 37 x 37 x 64 or 64 new images from one image with a size of 37 x 37 pixels.

The third layer of convolution produces an output of 32 new images with a size of 18 x 18 pixels. At this layer, the image will appear blurred but have meaningful features on the input image.

The fourth convolution layer will receive input in the form of 32 images with a size of 18 x 18 pixels and produce 16 images measuring 9 x 9 pixels. In this layer, the image will not look like a typical brain image because it has been extracted repeatedly in the previous layer.

The fifth convolutional layer which is the last layer in the block convolutional will have a unique feature on the input image that comes from the output of the previous layer. Although the resulting image does not have a brain-like shape, the unique feature will be entered into the neuron / dense layer to calculate the weight of each hidden layer neuron.

This layer will generate 8 images of 4 x 4 pixels, which is too small to be interpreted by the human eye, but rich in unique features of the input image.

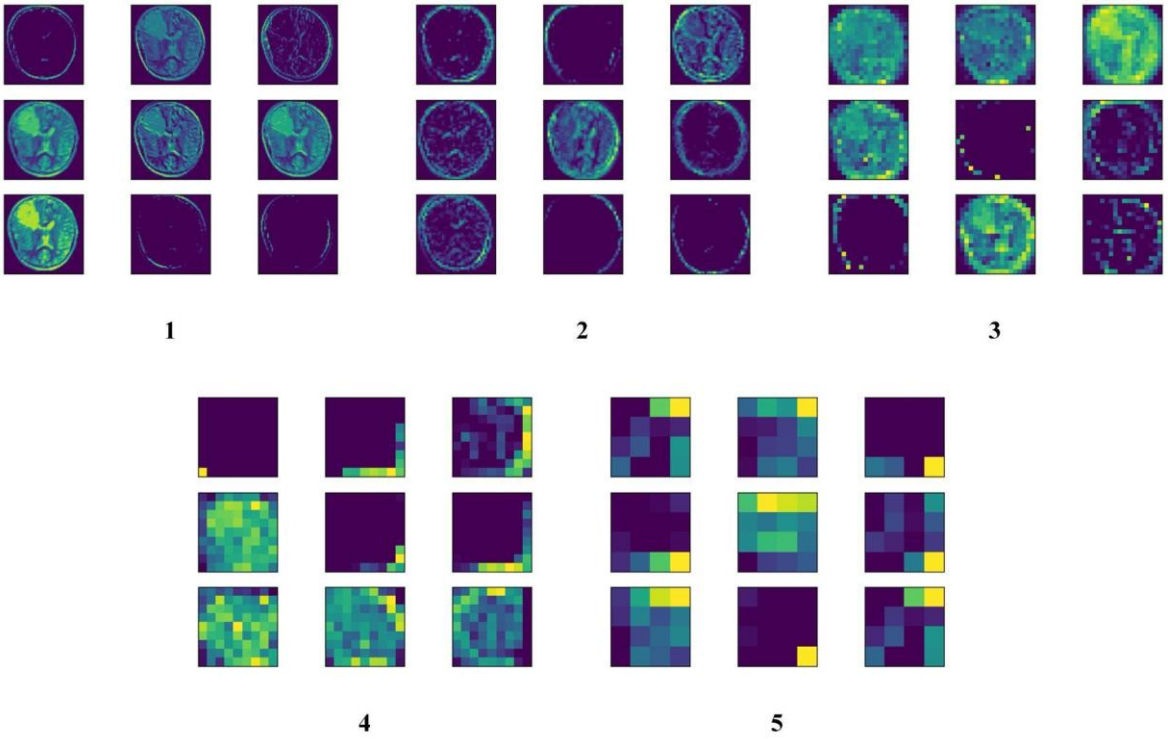

Fig. 12. Convolutional Result

The last layer of the model is the weighting layer of the artificial neural network. Before the output of the last layer of convolution is fed into the neurons, the input will be converted from a multidimensional array / multidimensional tensor to a one-dimensional array/tensor through the global average pooling layer. Then after that, the input can calculate the weights on each neuron or perceptron to more easily reach the convergence value and predict it in the output layer.

Deep learning modeling often results in models that are overfitting, underfiting or even models that do not have optimal hyperparameters to get the maximum value. The problem of determining the optimal model with maximum results can take a long time and waste resources if the goal is not achieved. Therefore, the hyperparameter tuning method is considered to be able to overcome these problems.

The hyperparameter tuning method works by testing each initialized scenario and running it with relatively short iterations, then monitoring the change in values. If there is no significant change in the model over a short iteration span, the model will be tested with different hyperparameter values.

The application of hyperparameter tuning to the proposed model results in a model with optimal and maximum values that can be seen in Figures 4.18 and 4.19 and the best learning rate is (50178 x 10-8).

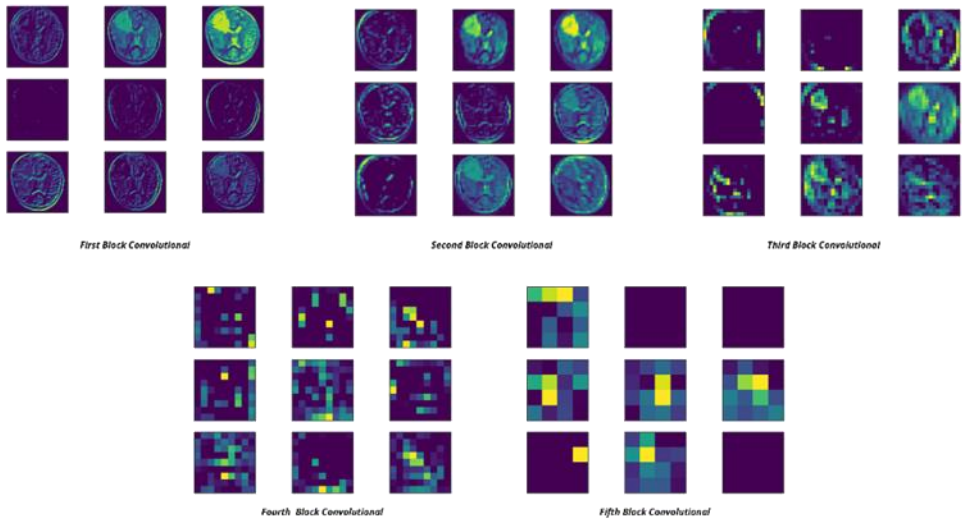

Fig. 13. Convolution Result of Hypermodel

Based on Figure 13, the convolution results can extract and segment the tumor part more accurately than the previous model. Provides more meaningful and rich features for the model run.

The trained model will be evaluated based on its prediction results on data that has never been seen in the testing set. The prediction results are assessed based on loss, accuracy, sensitivity, specificity, precision, F1-Score, and dice similarity coefficient (DSC) metrics.

In the process of training the model with 50 epochs/iterations, an accuracy of 96.6% was obtained with a loss of 12%. Despite getting a relatively high accuracy, the graph that tends to fluctuate indicates that the model has difficulty in reaching a convergent value or recognizing patterns in the image.

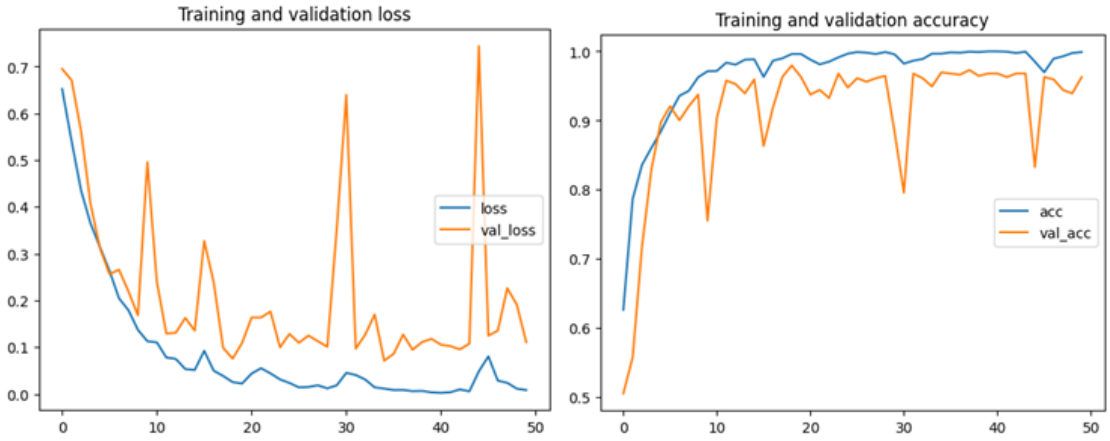

Fig. 14. Training and validation loss accuracy model

While the training process on the hypermodel with 50 epochs/iterations obtained an accuracy of 98.6% with a loss value of 5.8%. When viewed in Figure 14, the graph shows convergence after passing more than 30 epochs and indicates that the model has succeeded in recognizing patterns in the image better than the previous model and indicates that hyperparameter tuning has succeeded in improving model performance with the most optimal hyperparameter value with maximum results.

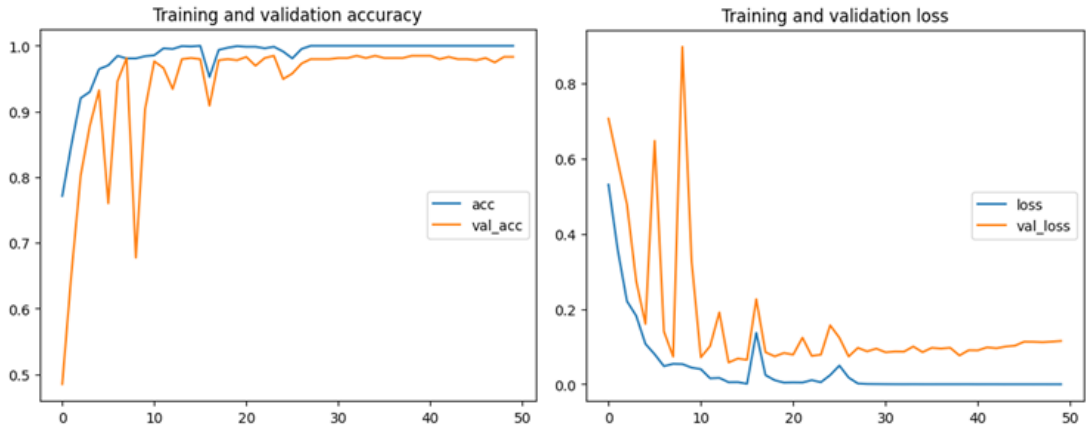

Fig. 15. Training and validation loss accuracy hypermodel

The performance of the model using the metrics of sensitivity, specificity, precision, F1-Score and dice similarity coefficient provides quite good performance with results above 90%.

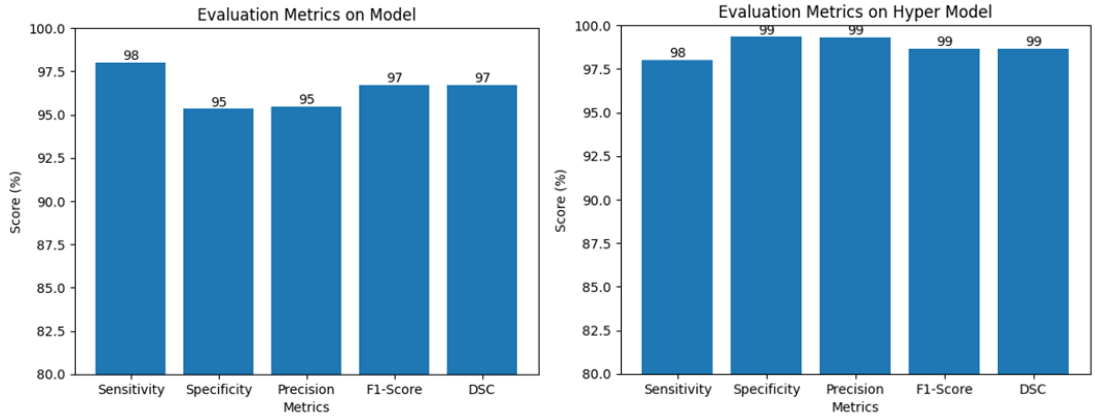

Fig. 16. Model and hypermodel performance metrics

Based on the performance results of the model and hypermodel on the metrics of sensitivity, specificity, precision, F1-Score, and dice similarity coefficient, the hypermodel has the highest value so that the hypermodel is chosen as the model used for the brain tumor prediction test process on magnetic resonance imaging scans.

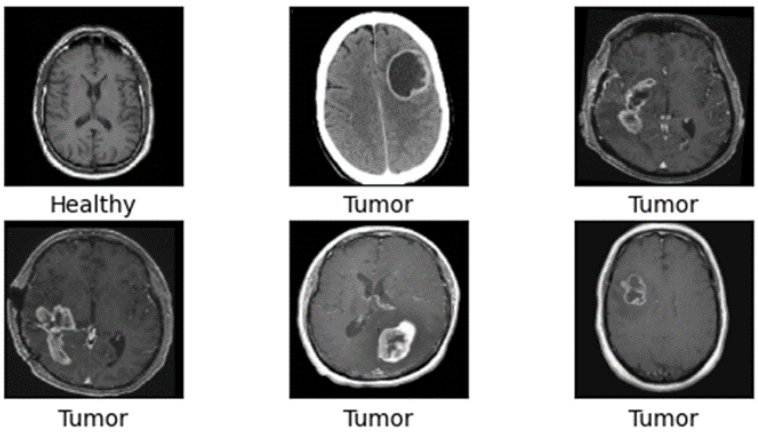

Fig. 17. Hypermodel prediction results

Prediction of brain tumors on magnetic resonance images with deep neural network methods provides a fairly good performance in determining whether the brain is diagnosed with a tumor or is in a healthy state. This is based on the accuracy, sensitivity, specificity, precision, F1-score, and dice similarity coefficient metric values that achieve performance above 95%, where each metric has a

value of 98% accuracy, 98% sensitivity, 99.3% specificity, 99.3% precision, 98.6% F1-score, and 98.6% dice similarity coefficient.

Some of these metrics have even exceeded the performance of the model by Isselmou Abd El Kader, Guizhi Xu, Zhang Shuai, Sani Saminu, Imran Javid, Isah Salim Ahmad in their research using the deep wavelet auto encoder method with a similar case, namely brain tumor prediction on MRI images [5]. The interpretation of each metric used as an evaluation of the deep learning model used to predict tumors and its comparison to research [5] is as follows.

The accuracy value obtained by the hypermodel reached 98.6% on the test set. The total test set used was 300 samples, which means that out of 300 samples including tumor and healthy samples, 294 samples were predicted/classified by the model correctly and 6 of them were predicted incorrectly. When compared with research [5], the accuracy performance can be seen in Figure 18.

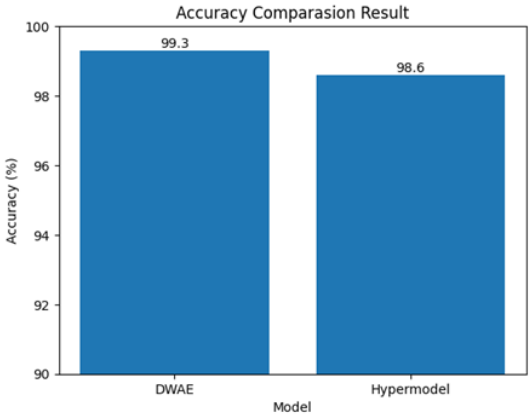

Fig. 18. Comparison of hypermodel accuracy

The sensitivity value obtained by the hypermodel reaches 98% on the tumor test set. Out of 150 brain-diagnosed samples, the hypermodel successfully predicted 147 samples correctly and 3 of them were predicted incorrectly. When compared to research [5] which only achieved a sensitivity performance value of 95.6%, hypermodel has a relatively higher value.

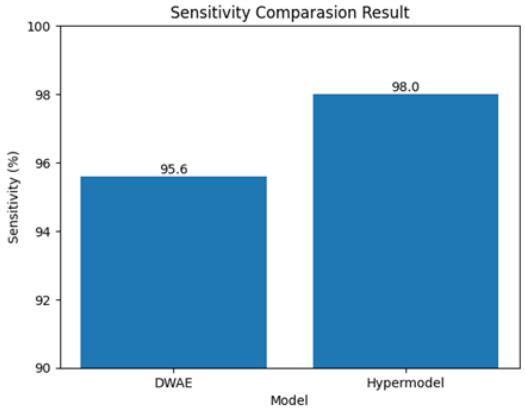

Fig. 19. Comparison of sensitivity

The specificity value obtained by the hypermodel reaches 99.3% on the tumor test set. Out of 150 brain-diagnosed samples, the hypermodel successfully predicted 149 samples correctly and 1 of them was predicted incorrectly. When compared to research [5] which only achieved a specificity performance value of 95.6%, hypermodel has a relatively higher value.

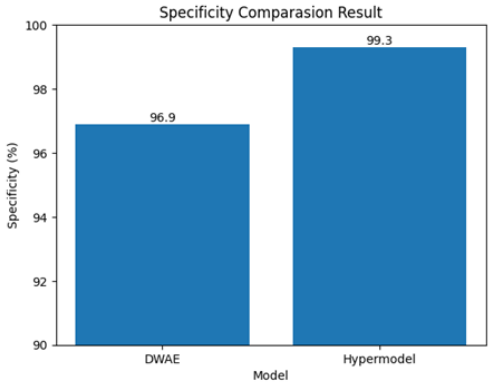

Fig. 20. Comparison of specificity

Precision obtained by the hypermodel reached 99.3% on the test set. This means that out of 150 samples that were predicted to have tumors, the model successfully predicted 149 samples of tumors correctly and 1 sample was wrongly predicted. When compared to research [5] which only achieved a precision performance value of 97.4%, hypermodel has a relatively higher value.

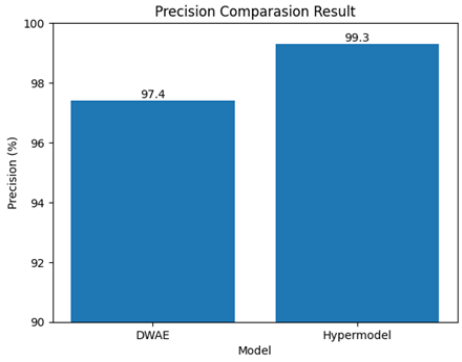

Fig. 21. Comparison of precision

The F1-score obtained by the hypermodel is 0.986 where the maximum value is 1. From the 300 sample test set the model achieved an equilibrium value of 0.986 in predicting tumors correctly and true negatives.

The dice similarity coefficient obtained by the hypermodel is 98.6%. Out of 300 actual test set samples, the model's predicted value is close to the actual value of 296 samples. When compared to research [5] which has a dice similarity performance of 96.5%, hypermodel achieves a higher similarity value.

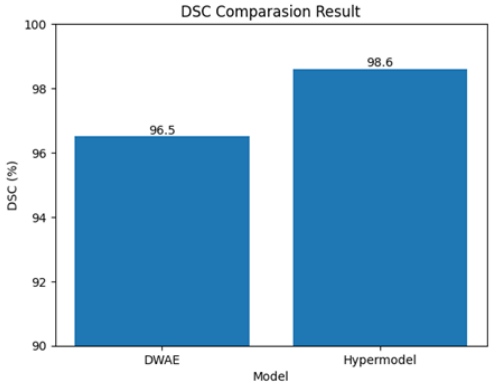

Fig. 22. DSC Comparison

4. Conclusion

Based on the results of research and discussion, it can be concluded that the hypermodel obtained managed to achieve performance above the model [5] on the metrics of sensitivity, specificity, precision, and DSC with values of 98.6%, 98%, 98%, 98.3% respectively. Although the accuracy value is only 98.3% below the model [5] with an accuracy of 99.3%, the hypermodel managed to achieve better performance on the other four metrics with a prediction speed of 0.2 seconds.

The hypermodel performance is achieved with the configuration of learning rate (50178 x 10-8), five block convolutional layers with each number of filters 64, 160, 64, 32, and 224, dense layer with 72 neurons/perceptron and a drop-out value of 0.2. The hyperparameter value is obtained from the hyperparameter tuning process to improve the initial model into a hypermodel. Thus, the hypermodel successfully achieved convergence on the dataset in detecting human brain tumors.

Acknowledgment

The authors would like to acknowledge the Faculty of Computer Science, Institut Informatika dan Bisnis Darmajaya, Lampung, Indonesia for providing financial support.

Declarations

Author contribution. The contribution or credit of the author must be stated in this section.
Conflict of interest. The authors declare no conflict of interest.
Additional information. No additional information is available for this paper.

References

[1] World Health Organization, *Guide to cancer early diagnosis*. Geneva: World Health Organization, 2017. Accessed: Dec. 01, 2023. [Online]. Available: https://iris.who.int/handle/10665/254500

[2] 'Cancer'. Accessed: Dec. 01, 2023. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/cancer

[3] R. A. Ramadhani, B. W. Pangestu, and R. Purbaningtyas, 'BRAIN TUMOR CLASSIFICATION USING CONVOLUTIONAL NEURAL NETWORK WITH EFFICIENTNET-B3 ARCHITECTURE', *Just IT J. System. Inf. Technol. Inf. And Comput.*, vol. 12, no. 3, Art. no. 3, Aug. 2022, doi: 10.24853/justit.12.3.55-59.

[4] S. Heranurweni, B. Destyningtias, and A. Kurniawan Nugroho, 'IMAGE POLA CLASSIFICATION IN BRAIN TUMOR PATIENTS BASED ON A CIRCLE SYNTHESIS NETWORK (CASE STUDY OF EMERGENCY HANDLING OF BRAIN TUMOR PATIENTS)', *Elektrika*, vol. 10, no. 2, p. 37, Oct. 2018, doi: 10.26623/elektrika.v10i2.1169.

[5] I. Abd El Kader *et al.*, 'Brain Tumor Detection and Classification on MR Images by a Deep Wavelet Auto-Encoder Model', *Diagnostics*, vol. 11, no. 9, p. 1589, Aug. 2021, doi: 10.3390/diagnostics11091589.

[6] A. Saleh, R. Sukaik, and S. S. Abu-Naser, 'Brain Tumor Classification Using Deep Learning', in *2020 International Conference on Assistive and Rehabilitation Technologies (iCareTech)*, Gaza, Palestine: IEEE, Aug. 2020, pp. 131-136. doi: 10.1109/iCareTech49914.2020.00032.

[7]A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: concepts, tools, and techniques to build intelligent systems.*Second edition. in Covid-19 collection. Beijing Boston Farnham Sebastopol Tokyo: O'Reilly, 2019.

[8] F. Chollet, *Deep learning with Python*, Second edition. Shelter Island: Manning Publications, 2021.

[9] S. Albawi, T. A. Mohammed, and S. Al-Zawi, 'Understanding of a convolutional neural network', in *2017 International Conference on Engineering and Technology (ICET)*, Antalya: IEEE, Aug. 2017, pp. 1-6. doi: 10.1109/ICEngTechnol.2017.8308186.

[10] M. A. Nielsen, 'Neural Networks and Deep Learning', 2015, Accessed: Dec. 04, 2023. [Online]. Available: http://neuralnetworksanddeeplearning.com

[11]A. M. Barhoom, S. Abu-Naser, E. Alajrami, B. Abu-Nasser, M. Musleh, and A. Khalil, 'Handwritten Signature Verification using Deep Learning', Dec. 2019.

[12] S. A. Hicks *et al.*, 'On evaluation metrics for medical applications of artificial intelligence', *Sci. Rep.*, vol. 12, no. 1, p. 5979, Apr. 2022, doi: 10.1038/s41598-022-09954-8.

[13]R. Parikh, A. Mathai, S. Parikh, G. Chandra Sekhar, and R. Thomas, 'Understanding and using sensitivity, specificity and predictive values', *Indian J. Ophthalmol.*, vol. 56, no. 1, p. 45, 2008, doi: 10.4103/0301-4738.37595.