Accredited Ranking SINTA 2
Decree of the Director General of Higher Education, Research, and Technology, No. 158/E/KPT/2021
Validity period from Volume 5 Number 2 of 2021 to Volume 10 Number 1 of 2026

Published online on: **http://jurnal.iaii.or.id**

JURNAL RESTI
(Rekayasa Sistem dan Teknologi Informasi)
Vol. 6 No. 4 (2022) 538 - 547 ISSN Media Electronic: 2580-0760

Classification of Brain Tumors on MRI Images Using Convolutional Neural Network Model EfficientNet

Muhammad Aji Purnama Wibowo[1], Muhammad Bima Al Fayyadl[2], Yufis Azhar[3], Zamah Sari[4]
[1,2,3,4]Informatika, Teknik, Universitas Muhammadiyah Malang
[1]ajipurnama@webmail.umm.ac.id, [2]muhammadbimaalfayyadl@webmail.umm.ac.id, [3]yufis@umm.ac.id, [4]zamahsari@umm.ac.id

Abstrak

A brain tumor is a lump caused by an imperfect cell turnover cycle in the brain and can affect all ages. Brain tumors have 4 grades, namely grades 1 to 2 are benign tumor grades, and grades 3 to 4 are malignant tumor grades. Therefore, early identification of brain tumor disease is very important in providing appropriate treatment and treatment. This study uses a dataset obtained through the Kaggle website titled Brain Tumor Classification (MRI). The number of data is 3264 images with details of Glioma tumors (926 images), Meningioma tumors (937 images), pituitary tumors (901 images), and without tumors (500 images). In this study, there are 4 scenarios with different testers. This study proposes the classification of brain tumors using Hyperparameter Tuning and EfficientNet models on MRI images. The EfficientNet model used is the EfficientNetB0 and EfficientNetB7 models with the architecture used are the input layer, GlobalAveragePooling2D layer, dropout layer, and dense layer as well as adding augmentation data to the dataset to manipulate the data in order to improve the results of the proposed model. After building the model, the results of accuracy, precision, recall, and f1-score will be obtained in each scenario. Accuracy results in Scenario 1 are 91%, scenario 2 is 95% accurate, scenario 3 is 95%, and scenario 4 is 98%.

Keywords: Brain Tumor, EfficientNet, Convolutional Neural Network, MRI

1. Introduction

The brain is a very important part of the human body. The brain has a role in controlling and regulating human reactions, such as providing the ability to think, feel, and remember things that make us human. The brain has more than 200 million nerve cells including the cerebral cortex and contains about 16.3 billion neurons that are responsible for controlling all human activities. The cerebral cortex is the outer part of the human brain that functions to process sensory performance, produce motor activity, and high-level cognitive activities [3]. However, this cerebral cortex cannot function properly if there is a disorder that can be called a disease.

The most common disease in the brain is a brain tumor. This disease can occur at all ages, especially in the elderly, and children are no exception [4]. A brain tumor is a lump caused by an imperfect cell turnover cycle in the brain. The existence of this non-functioning cell should have died but was still trying to live. This causes the accumulation of excess cells, resulting in the formation of the lump. Brain tumors are characterized by the growth of abnormal cells in or around the brain. Furthermore, these abnormal cells grow unnaturally and uncontrollably in the brain [5].

In general, brain tumors are divided into two types, namely benign tumors, and malignant tumors [1], the tumor grade range is 1 to 4. Benign tumors based on the rate of growth and the way of spreading are in grades 1 to 2 and do not have the potential to become malignant tumors [6]. Meanwhile, malignant tumors based on the rate of growth rate and the way it spreads are at levels 3 to 4. Therefore, malignant tumors are very dangerous because they can spread to other cells and cause irreversible damage to brain cells [2]. When benign or malignant tumors grow, they can cause increased pressure damage to the brain within the human skull. This phenomenon can be the cause of the threat to a person's life which ends in death.

Accepted: 02-06-2022 | Received in revised: 18-07-2022 | Published: 22-07-2022

Based on data from the Indonesian Ministry of Health, the Indonesian Health Profile in 2018 that malignant tumors are the second type of disease that causes death after cardiovascular (heart/blood vessel disorders). Meanwhile, at the world level, statistics recorded 1/6 of deaths due to tumor disease. In 2018, 9.6 million deaths and 70% were in developing countries, including Indonesia [7].

In the recent lustrum, the majority of medical personnel in diagnosing brain tumor disease against a patient involve technology as a supporting media for examination. Meanwhile, a minority of other medical personnel use manual diagnosis. The technology in question is Magnetic Resonance Imaging (MRI) image examination [8]. MRI is a set of scans using a magnetic field and a computer to capture images of the brain on film [9]. Tumors can be detected by examining the structures in the brain [10]. MRI is considered the best way to make a diagnosis because of its very high sensitivity and accuracy [11]. This is the reason that medical personnel use MRI images.

Many research sources provide information regarding the workings of MRI images regarding the classification of brain tumors in patients with them. The way this MRI image works is to classify the categories of benign and malignant tumors, namely through a classification system. The classification system in question is machine learning and deep learning methods. The machine learning method is a machine development that is able to learn by itself without any direction from the user. While deep learning is one part of Machine Learning that works by imitating the nervous system based on an active human brain. This artificial system is called Artificial Neural Networks or often abbreviated as ANN. The deep learning algorithms that are most often used are Convolution Neural Network, LSTM, and RNN. Another benefit of applying machine learning to brain tumors is the accuracy of the system, which can help doctors identify tumors on the patient's head. The most commonly used machine learning methods today are Naive Bayes, Decision Tree, KNN, SVM, and Random Forest.

Previous research related to brain tumor image classification has been carried out by Agus Eko and his colleagues (2021) with the title "Convolutional neural network with hyperparameter tuning for brain tumor classification" proposing the application of the CNN method combined with Hyperparameter Tuning to find the best parameter values. The researcher made 3 model test scenarios. Scenario 1 uses the proposed CNN model design, scenario 2 uses the second best result from hyperparameter tuning, and scenario 3 uses the best result from hyperparameter tuning. Among the three scenarios, scenario 3 is the best with an accuracy of 96% [1].

A previous study related to the classification of brain tumor images has been carried out by N. Abiwinanda and his colleagues (2018) with the title "Brain Tumor Classification Using Convolutional Neural Networks" proposing the application of the CNN algorithm for the classification of brain tumors. Using a dataset consisting of glioma (708 images), meningioma (1426 images), and pituitary (930 images). The model that was built obtained an accuracy of 94.68% [12].

Previous research related to the classification of brain tumor images has been carried out by Abdu Gumaei and his colleagues (2019) with the title "A Hybrid Feature Extraction Method with Regularized Extreme Learning Machine for Brain Tumor Classification" proposing the Regularized Extreme Learning Machine (RELM) method in classifying types of brain tumors. brain tumor. The datasets used were meningiomas (1426 images), gliomas (708 images), and pituitary (930 images). The proposed model produces an accuracy of 94.23% [13].

Previous research related to the classification of brain tumor images has been carried out by A. Yang and several colleagues (2019) with the title "Research on Feature Extraction of Tumor Image Based on Convolutional Neural Network" proposed that when conducting a tumor classification system, researchers used the Convolutional Neural Network method. by applying two convolutional models, namely the Xception model and the Dense Net model which have been trained previously to improve the accuracy of the Convolutional Neural Network algorithm [14].

Previous research related to the classification of brain tumor images has been carried out by Hassan Ali Khan and several colleagues (2020) with the title "Brain tumor classification in MRI image using convolutional neural network" using a transfer learning approach to perform comparisons, which compare the performance of the model. Pre-trained VGG-16, ResNet-50, and Inception-v3. After testing the dataset, the results of the model that the researcher trained showed that the accuracy of the model being trained was very effective and had a very low level of complexity, achieving 100% accuracy, while VGG-16 reached 96%, ResNet-50 reached 89% and Inception- V3 achieves 75% accuracy [6].

Previous research related to the classification of brain tumor images has been carried out by Nayak and several colleagues (2022). With the title "Brain Tumor Classification Using Dense Efficient-Net" by classifying Meningioma, Glioma, and pituitary MRI images using the Dense EfficientNet method. This method process has implemented a pre-trained model from EfficientNet by adding dense and dropout layers and using min-max normalization The accuracy obtained with the EfficientNet model is 98.78% [15].

Based on the research above, the purpose of this study is to exceed the accuracy results of previous studies. The method proposed in this study is to apply the EfficientNet model. The EfficientNet models used in this study are the EfficientNetB0 and EfficientNetB7 models. Both of these models have better accuracy and efficiency than other pre-trained models. It is hoped that these two models will get higher accuracy results than previous studies and the hope is that they can carry out the process of classifying brain tumors more precisely.

2. Research methods

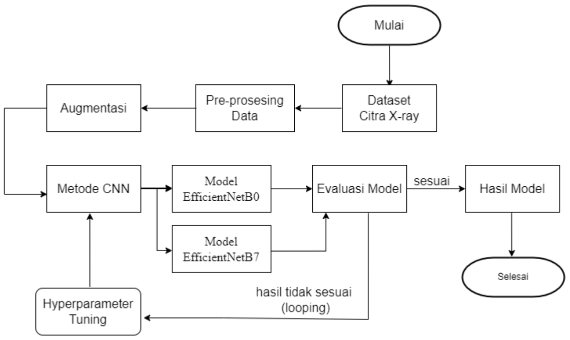


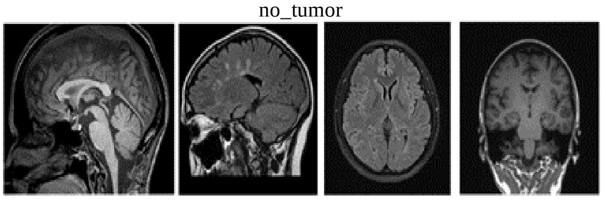
Figure 1. Research method flowchart

Figure 1 illustrates the research flow from beginning to end. Starting with retrieving the dataset from the Kaggle website. Then carried out several stages of data preprocessing in the form of data splitting by dividing the "Brain Tumor Classification (MRI)" dataset into 3 folders consisting of train, validation, and test data folders. The next stage after preprocessing is to process the results of grouping the data. In the data train, the modeling stage is carried out which is used to train the data. Next, perform data augmentation. After that, the next process is making and training the model. In making the model, the hyperparameter tuning technique was added to the EfficientNetB0 and EfficientNetB7 models to obtain optimal parameters for the proposed model. The last part of this research is the model evaluation process.

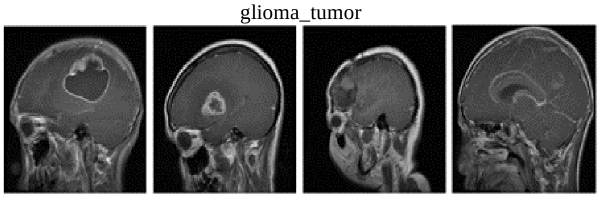
2.1. Dataset

The dataset used in this study was obtained from the Kaggle website titled "Brain Tumor Classification (MRI)". This dataset has 4 classes in which the total number of data is 3264 images with details of glioma tumors (926 images), meningioma tumors (937 images), without tumors (500 images), and pituitary tumors (901 images) [16]. The following is an example of an image of a tumor in each class as shown in Figure 2.

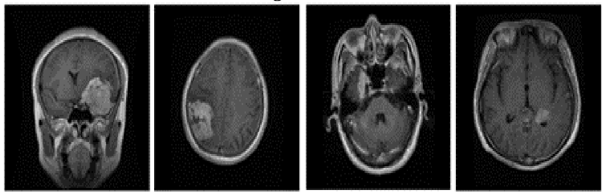
no_tumor



glioma_tumor



meningioma_tumor



pituitary_tumor

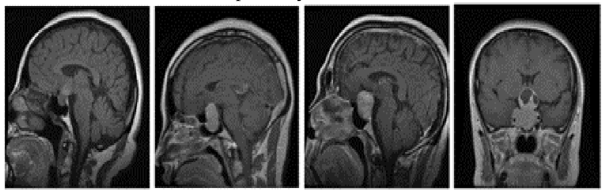


Figure 2. Sample images on each brain tumor label

2.2. Preprocessing Data

The initial stage of research on this dataset is to collect data. Data collection was obtained from the dataset provider site, namely Kaggle. The dataset used is called "Brain Tumor Classification (MRI)". The second stage performs preprocessing on the dataset such as data splitting, namely to separate or divide the data into train, test, and validation data. Train and validation data are used when conducting model training and test data are used when predicting unseen data using trained data. The Brain Tumor Classification MRI dataset will split the defined data with a ratio of 80% train data, 10% validation data, and 10% test data. Furthermore, Augmentation conducts training on the data where the data being trained is train data and validation data. Data augmentation is done using ImageDataGenerator from the Keras library. Furthermore, training data is carried out to train with the CNN model to be trained.

2.3. Augmentation

At this stage, the data augmentation process is carried out which serves to increase the variance of the image so that it can help increase the accuracy of the model [17]. The augmentation parameters used in this study are rotation_range = 30, zoom_range = 0.2, width_shift_range = 0.1, height_shift_range = 0.1, horizontal_flip = True, vertical_flip = False, and rescale = 1. / 255.

2.4. Model Architecture

This study builds on the architectural design of the CNN model to classify brain tumors on MRI images. In the modeling process, the hyperparameter tuning process is added to find the model with the best parameters, so that the proposed model can achieve good performance [1]. The model proposed in this study uses pre-trained models from EfficientNet, namely EfficientNetB0 and EfficientNetB7. The EfficientNet model has better accuracy and efficiency compared to other pre-trained models. EfficientNetB0 has 230 layers [15], while EfficientNetB7 has 813 layers. The model in this study includes the input layer (128x128) from the pre-trained model EfficientNet, GlobalAveragePooling2D, Dropout (0.2, 0.5) layer, and dense layer (128, 512, and 1024). The pooling layer is the division of the feature map of an image into several sub-sections and reducing the parts in new sub-sections. The MaxPool layer will take the maximum value from the feature map, while the AveragePool will take the average value from the feature map to be able to retrieve all image information [21]. Use the GlobalAveragePooling2D layer here to select the average feature value of an image. The difference between AveragePooling2D and GlobalAveragePooling2D is that the pool value of AveragePooling2D is determined from the feature map while the GlobalAveragePooling2D pool value takes the value from the feature map. The results of the architectural design of the model are illustrated in Table 1.

Table 1. CNN Model Architectural Design

Layer	Filter	Kernel Size	Activation
EfficientNet_Input (128, 128)	-	-	-
GlobalAveragePooling2D	-	-	-
Dropout	0.2, 0.5	-	-
Dense	512, 1024	-	relu
Dropout	0.2, 0.5	-	-
Dense	128	-	relu
Dense	4	-	softmax

2.5. Hyperparameter Tuning

Hyperparameter Tuning provides an optimal parameter search method for the proposed Convolutional Neural. CNN network model combined with Hyperparameter Tuning requires parameter settings, such as kernel size, step, number of channels, and number of dropouts [18]. The hyperparameter setting gives the best combination of parameters to the model that shows maximum results [19]. In this study, Hyperparameter Tuning aims to provide the best parameter values through the proposed comparison parameters. The comparative parameters are illustrated in Table 2 as follows.

Table 2. Comparative parameters in hyperparameter tuning

Parameter	Comparative Value
Dropout	0.2, 0.5
Dense Layer	1024, 512
Optimizer	Adam, Adamax, RMSprop

3. Results and Discussions

At this stage, it is part of getting the results of the architectural model that has been built in this study in the case of tumor disease classification.

The first step in this study was taking data from the sites mentioned in the research methods section and splitting the data with a ratio of 80% training data, 10% validation data, and 10% test data. Next, perform the augmentation process on the training data. The parameters used in the augmentation process are the augmentation parameters used in this study are rotation_range = 30, zoom_range = 0.2, width_shift_range = 0.1, height_shift_range = 0.1, horizontal_flip = True, vertical_flip = False, and rescale = 1. / 255.

Next, apply a callback technique by utilizing ModelCheckpoint to monitor model performance in real-time, where when the performance in the latest epoch is much worse than the previous epoch, the results of the latest epoch will not be saved. This technique will only save the best model of all epochs so that it will reduce memory usage [22]. In this study, there were 4 model scenarios that were tested with different parameters. The following table presents model testing scenarios and a description of each model, which can be seen in Table 3.

Table 3. Description of the model scenario

Scenario	Description
Model 1	Model *EfficientNetB0*
Model 2	Model *EfficientNetB7*
Model 3	EfficientNetB0 model with best hyperparameter results
Model 4	EfficientNetB7 model with best hyperparameter results

3.1 Model 1 Scenario Testing

The model 1 scenario test uses the proposed model, namely the EfficientNetB0 model. In the model scenario using the parameters in Table 4.

Table 4. Scenario architecture model 1

Layer	Filter	Kernel Size	Activation
EfficientNet_Input (128, 128)	-	-	-
GlobalAveragePooling2D	-	-	-
Dropout	0.2	-	-
Dense	1024	-	relu
Dropout	0.2	-	-
Dense	128	-	relu
Dense	4	-	softmax

The model architecture used in scenario 1 has been described in Table 4, with details Dropout = 0.2, Dense = 1024 with relu activation, Dropout = 0.2, Dense = 128 with relu activation, Dense = 4 with softmax activation and using the Adam Optimizer, learning rate 0.00146, the epoch is 100, and uses the categorical_crossentropy classification.

After testing, the next step is to create a plot to display all the results of the training process. The results are

DOI: https://doi.org/10.29207/resti.v6i4.4119

depicted in the form of a line graph. This plotting is useful to see if there is an improvement from each iteration or not. Graphs can also be used to see whether the results of the model made are overfitting or not. The following are the results of plotting on the EfficientNetB0 model, which can be seen in Figure 3 and Figure 4.

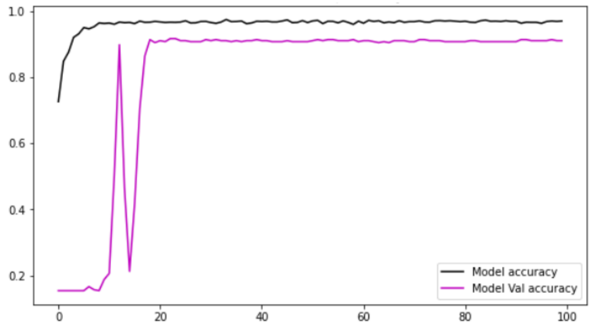


Figure 3. The results of the plot of the accuracy of the model 1 scenario

Figure 3 shows the results of the accuracy plot from model scenario 1. In the graph, the initial value of the validation accuracy from epochs 0 to 20 experienced unstable graph movements, but when epochs 21 to 100 the graph movement became stable approaching number 1. The movement of accuracy was unstable. at the beginning of the epoch because the model in this scenario is still in the dataset learning stage so the movement of accuracy cannot be stable, so at the 21st to 100th epoch, the model can be stable because it has learned the dataset.

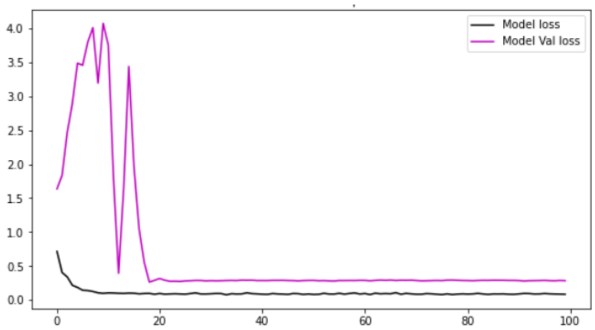


Figure 4. The results of the plot loss scenario model 1

Figure 4 shows the loss plot results from scenario model 1. In the graph the initial value of the validation loss from epochs 0 to 20 experienced unstable graph movements, but at epochs 21 to 100, the graph movement became stable near 0. Unstable loss movements at the beginning of the epoch because the model in this scenario is still in the dataset learning stage so the loss movement cannot be stable, so that at the 21st to 100th epoch the model can be stable because it has learned the dataset.

After getting the graphic results from the training that has been done, the next step is to evaluate the models that have been built. The performance details are then visualized in the form of a classification report through Table 5 as follows:

Table 5. classification report scenario model 1

Classification Report	
Accuracy	91 %
Precision	93 %
Recall	92 %
F1-Score	92 %

The results of the model architecture in the model 1 scenario show 91% accuracy, 93% precision, 92% recall, and 92% f1-Score.

Furthermore, the evaluation of the model can also be seen through the confusion matrix table to measure the performance of the machine learning method in knowing how much the model is able to predict correctly and incorrectly from the total data [20]. The results of the confusion matrix test scenario model 1 can be seen in Figure 5.

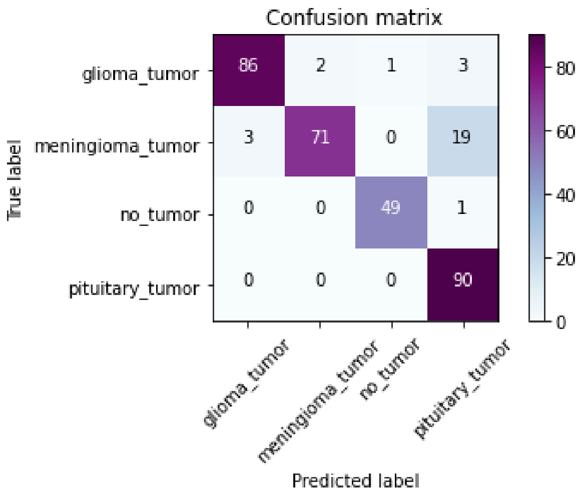


Figure 5. Confusion matrix scenario model 1

In Figure 5 is the result of the confusion matrix model 1, it can be concluded that in the glioma tumor class there are 86 image data that are predicted to be correct and 3 image data are predicted to be incorrect, in the meningioma tumor class there are 71 image data that are correctly predicted and 2 image data are correct. predicted incorrectly, in class no tumor there were 49 image data that were predicted to be correct and 1 image data was predicted to be incorrect, and in the pituitary tumor class there were 90 image data that were predicted correctly and 23 image data that were predicted incorrectly.

3.2 Model 2 Scenario Testing

The model 2 scenario test uses the proposed model, namely the EfficientNetB7 model. In the model scenario using the parameters in Table 6.

The model architecture used in scenario 2 has been described in Table 6, with details Dropout = 0.2, Dense = 1024 with relu activation, Dropout = 0.2, Dense = 128 with relu activation, Dense = 4 with softmax activation and using the rmsprop optimizer, learning rate 0.00146, the epoch is 100, and uses the categorical_crossentropy classification.

Table 6. Scenario architecture model 2

Layer	Filter	Kernel Size	Activation
EfficientNet_Input (128, 128)	-	-	-
GlobalAveragePooling2D	-	-	-
Dropout	0.2	-	-
Dense	1024	-	relu
Dropout	0.2	-	-
Dense	128	-	relu
Dense	4	-	softmax

After conducting the training, the next step is to make a plot to display all the results of the training process. The results are depicted in the form of a line graph. This plotting is useful to see if there is an improvement from each iteration or not. Graphs can also be used to see whether the results of the model made are overfitting or not. The following are the results of plotting without using hyperparameters in the EfficientNetB7 model, which can be seen in Figure 6 and Figure 7.

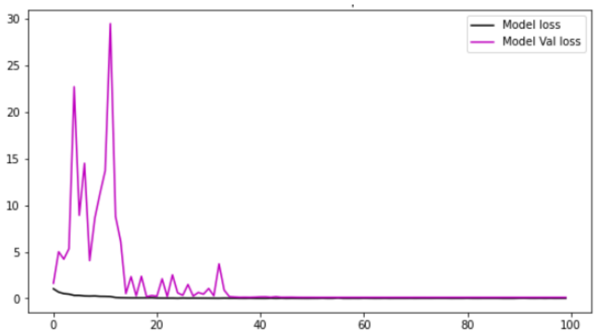


Figure 6. The results of the plot of the accuracy of the model 2 scenario

Figure 6 shows the results of the accuracy plot from model scenario 2. In the graph, the initial value of the validation accuracy from epochs 0 to 36 experienced unstable graph movements, but when epochs 37 to 100 the graph movement became stable approaching number 1. Unstable accuracy movements at the beginning of the epoch because the model in this scenario is still in the dataset learning stage so the movement of accuracy cannot be stable, so that at the 37th to 100th epoch the model can be stable because it has learned the dataset.

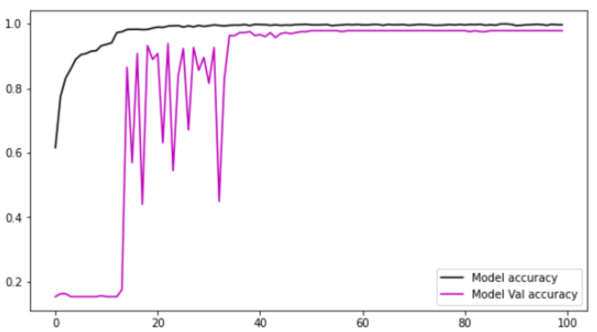


Figure 7. Results of plot loss for model 2 scenario

Figure 7 shows the results of the loss plot from scenario model 2. In the graph the initial value of the validation loss from epochs 0 to 36 experienced unstable graph movements, but at epochs 37 to 100, the graph movement became stable near 0. Unstable loss movements at the beginning of the epoch because the model in this scenario is still in the dataset learning stage so the loss movement cannot be stable, so that at the 37th to 100th epoch the model can be stable because it has learned the dataset.

After getting the graphic results from the training that has been done, the next step is to evaluate the models that have been built. The performance details are then visualized in the form of a classification report through Table 7.

Table 7. Results of classification report model 2

Classification Report	
Accuracy	95%
Precision	93 %
Recall	92 %
F1-Score	93 %

The results of the model architecture in the model 2 scenario show an accuracy value of 95%, 93% precision, 92% recall, and 93% f1-Score.

Furthermore, the evaluation of the model can also be seen through the confusion matrix table to measure the performance of the machine learning method in knowing how much the model is able to predict correctly and incorrectly from the total data [20]. The results of the confusion matrix test scenario model 2 can be seen in Figure 8.

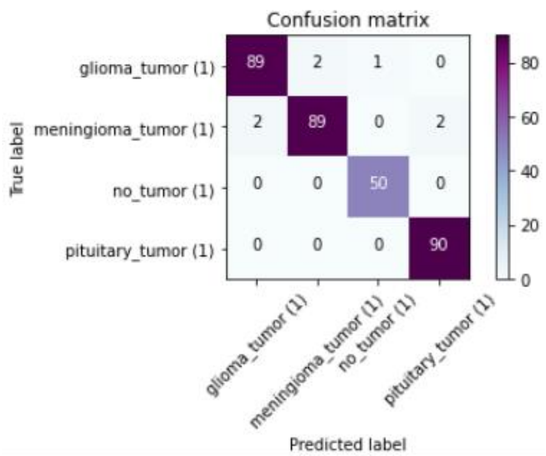


Figure 8. The results of the confusion matrix model 2

Figure 8 is the result of the confusion matrix model 2, it can be concluded that in the glioma tumor class there are 89 image data that are predicted to be correct and 2 image data are predicted to be incorrect, in the meningioma tumor class there are 89 image data that are predicted correctly and 2 image data are correct. predicted incorrectly, in the no tumor class there were 50 image data that were predicted to be correct and 1 image

DOI: https://doi.org/10.29207/resti.v6i4.4119

data was predicted to be incorrect, and in the pituitary tumor class there were 90 image data that were predicted to be correct and 2 image data were predicted to be incorrect. predicted incorrectly.

3.3 Model 3 Scenario Testing

The model 3 scenario test uses the proposed model, namely the EfficientNetB0 model with Hyperparameter tuning. The results of the Hyperparameter tuning with the EfficientNetB0 model are summarized in Table 8.

Table 8. Results of EfficientNetB0 hyperparameters tuning

Parameter	Optimizer	Dropout	Dense Layer	Akurasi
1	Adamax	0.2	1024	68%
2	Adam	0.2	1024	31%

From the results of the EfficientNetB0 Hyperparameter tuning in table 8, it can be concluded that parameter 1 is the best parameter because the accuracy obtained is 68%. In this scenario, the model uses the parameters from the results of the Hyperparameter tuning which are illustrated in Table 9.

Table 9. The architecture of the model 3 scenario

Layer	Filter	Kernel Size	Activation
EfficientNet_Input (128, 128)	-	-	-
GlobalAveragePooling2D	-	-	-
Dropout	0.2	-	-
Dense	1024	-	relu
Dropout	0.2	-	-
Dense	128	-	relu
Dense	4	-	softmax

The model architecture used in scenario 3 has been described in Table 9, with details Dropout = 0.2, Dense = 1024 with relu activation, Dropout = 0.2, Dense = 128 with relu activation, Dense = 4 with softmax activation and using the Adamax Optimizer, learning rate 0.00146, the epoch is 100, and uses the categorical_crossentropy classification.

After conducting the training, the next step is to make a plot to display all the results of the training process. The results are depicted in the form of a line graph. This plotting is useful to see if there is an improvement from each iteration or not. Graphs can also be used to see whether the results of the model made are overfitting or not. The following are the results of plotting using hyperparameters on the EfficientNetB0 model, which can be seen in Figure 9 and Figure 10.

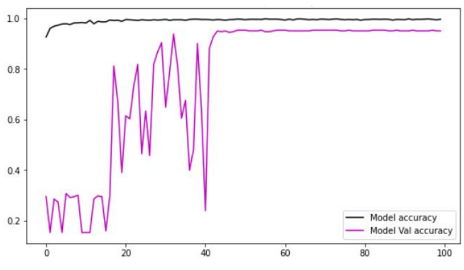


Figure 9. The results of the model 3 scenario accuracy plot

Figure 9 shows the results of the accuracy plot from model scenario 3. In the graph, the initial value of validation accuracy from epochs 0 to 40 experiences unstable graph movements, but when epochs 41 to 100 the graph movement becomes stable approaching number 1. The movement of accuracy is unstable at the beginning of the epoch because the model in this scenario is still in the dataset learning stage so the movement of accuracy cannot be stable, so that at the 41st to 100th epoch the model can be stable because it has learned the dataset.

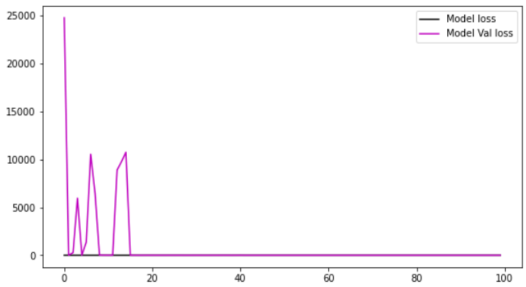


Figure 10. The results of the model 3 scenario accuracy plot

Figure 10 shows the results of the loss plot from the model 3 scenario. In the graph the initial value of the validation loss from epochs 0 to 17 experienced unstable graph movements, but at epochs 18 to 100, the graph movement became stable near 0. Unstable loss movements at the beginning of the epoch because the model in this scenario is still in the dataset learning stage so the loss movement cannot be stable, so that at the 18th to 100th epoch the model can be stable because it has learned the dataset.

After getting the graphic results from the training that has been done, the next step is to evaluate the models that have been built. The performance details are then visualized in the form of a classification report through Table 10.

Table 10. Result of classification report model 3

Classification Report	
Accuracy	95 %
Precision	95 %
Recall	95 %
F1-Score	95 %

The results of the model architecture in the model 3 scenario show an accuracy value of 95%, 95% precision, 95% recall, and 95% f1-Score.

Furthermore, the evaluation of the model can also be seen through the confusion matrix table to measure the performance of the machine learning method in knowing how much the model is able to predict correctly and incorrectly from the total data [20]. The results of the confusion matrix model 3 scenario test can be seen in Figure 11.

DOI: https://doi.org/10.29207/resti.v6i4.4119

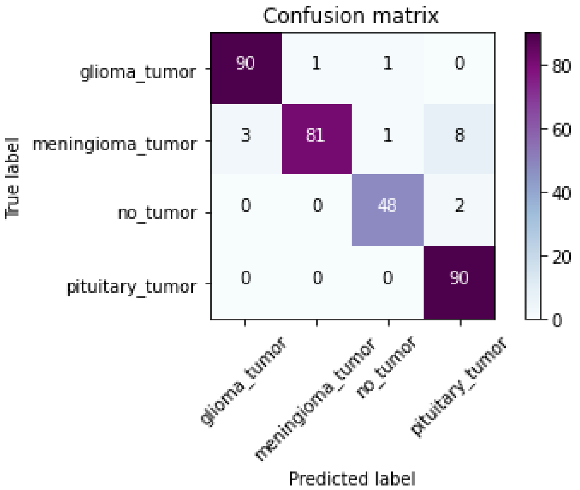


Figure 11. The results of the confusion matrix model 3

Figure 11 is the result of the confusion matrix model 3, it can be concluded that in the glioma tumor class there are 90 correct image data and 3 incorrectly predicted image data, in the meningioma tumor class there are 81 correctly predicted image data and 1 correct image data. predicted incorrectly, in class no tumor there were 48 image data that were predicted to be correct and 2 image data were predicted to be incorrect, and in the pituitary tumor class, there were 90 image data that were predicted correctly and 10 image data that were predicted incorrectly.

3.4 Model 4 Scenario Testing

The model 4 scenario test uses the proposed model, namely the EfficientNetB7 model with Hyperparameter tuning. The results of the Hyperparameter tuning with the EfficientNetB7 model are summarized in Table 11.

Table 11. Results of EfficientNetB7 hyperparameters tuning

Parameter	Optimizer	Dropout	Dense Layer	Akurasi
1	Adamax	0.2	1024	92%
2	Rmsprop	0.2	1024	71%

From the results of the EfficientNetB7 Hyperparameter tuning in table 11, it can be concluded that parameter 1 is the best parameter because the accuracy obtained is 92%. In this scenario, the model uses the parameters from the results of the Hyperparameter tuning which are illustrated in Table 12.

Table 12. CNN Model Architectural Design

Layer	Filter	Kernel Size	Activation
EfficientNet_Input (128, 128)	-	-	-
Global Average Pooling2D	-	-	-
Dropout	0.2	-	-
Dense	1024	-	relu
Dropout	0.2	-	-
Dense	128	-	relu
Dense	4	-	softmax

The model architecture used in scenario 4 has been described in Table 12, with details Dropout = 0.2, Dense = 1024 with relu activation, Dropout = 0.2, Dense = 128 with relu activation, Dense = 4 with softmax activation and using the Adamax Optimizer, learning rate 0.00146, the epoch is 100, and uses the categorical_crossentropy classification.

After conducting the training, the next step is to make a plot to display all the results of the training process. The results are depicted in the form of a line graph. This plotting is useful to see if there is an improvement from each iteration or not. Graphs can also be used to see whether the results of the model made are overfitting or not. The following are the results of plotting without using hyperparameters on the EfficientNetB7 model, which can be seen in Figure 12 and Figure 13.

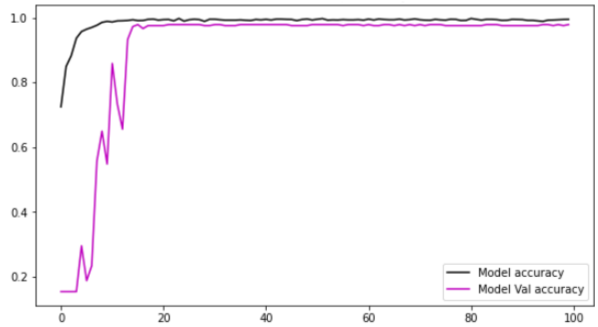


Figure 12. The results of the plot of the accuracy of the model 4 scenario

Figure 12 shows the results of the accuracy plot from model scenario 4. In the graph, the initial value of the validation accuracy from epochs 0 to 19 experienced unstable graph movements, but at epochs 20 to 100, the graph movement became stable approaching number 1. The movement of accuracy was unstable. at the beginning of the epoch because the model in this scenario is still in the dataset learning stage so the movement of accuracy cannot be stable, so at the 20th to 100th epoch, the model can be stable because it has learned the dataset.

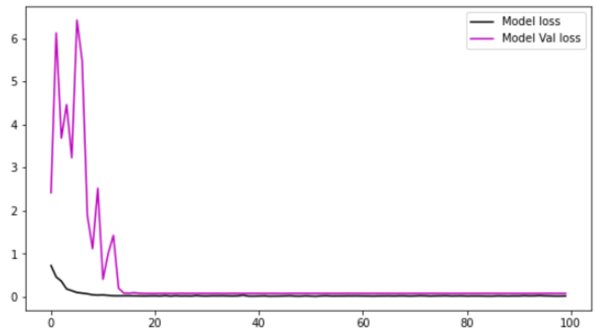


Figure 13. The results of the plot loss for the model 4 scenario

Figure 13 shows the loss plot results from model scenario 4. In the graph the initial value of the validation loss from epochs 0 to 15 experienced unstable graph movements, but at epochs 16 to 100, the graph movement became stable near 0. Unstable loss

DOI: https://doi.org/10.29207/resti.v6i4.4119

movements at the beginning of the epoch because the model in this scenario is still in the dataset learning stage so that the loss movement cannot be stable, so that at the 16th to 100th epoch the model can be stable because it has learned the dataset.

After getting the graphic results from the training that has been done, the next step is to evaluate the models that have been built. The performance details are then visualized in the form of a classification report through Table 13 as follows:

Table 13. Result of classification report model 4

Classification Report	
Accuracy	98 %
Precision	97 %
Recall	98 %
F1-Score	97 %

The results of the model architecture in the model 4 scenario show an accuracy value of 98%, 97% precision, 98% recall, and 97% f1-Score.

Furthermore, the evaluation of the model can also be seen through the confusion matrix table to measure the performance of the machine learning method in knowing how much the model is able to predict correctly and incorrectly from the total data [20]. The results of the confusion matrix test scenario model 4 can be seen in Figure 14.

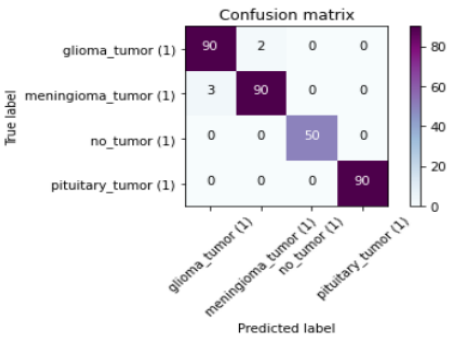


Figure 14. The results of the confusion matrix model 4

Figure 14 is the result of the confusion matrix model 4, it can be concluded that in the glioma tumor class there are 90 correct image data and 3 incorrectly predicted image data, in the meningioma tumor class there are 90 correctly predicted image data and 2 correct image data. predicted wrong, in the class no tumor there were 50 image data that were predicted to be correct and 0 image data were predicted to be wrong, and in the pituitary tumor class there were 90 image data that were predicted to be correct and 0 image data were predicted to be incorrect.

The test results of each model scenario show that there is a significant effect from the application of Hyperparameter Tuning in determining the best parameters for the proposed model in terms of accuracy. The results of the accuracy test in the four model scenarios are summarized in Table 14 as follows.

Table 14. Model Testing Results

Scenario	Accuracy	Precision	Recall	F1-Score
Model 1	91%	93%	92%	92%
Model 2	95%	93%	92%	93%
Model 3	95%	95%	95%	95%
Model 4	98%	97%	98%	97%

In this study, an evaluation of the model was carried out on previous studies [1]. Details from comparison to previous studies are illustrated in Table 15 as follows.

Table 15. Comparison table with previous research

Scenario	Dataset	Model	Accuracy
Agus Eko et al.[1]	Grade 4 Brain Tumor	CNN	96%
Model 1	Grade 4 Brain Tumor	*Efficient NetB0*	91%
Model 2	Grade 4 Brain Tumor	*Efficient NetB7*	95%
Model 3	Grade 4 Brain Tumor	*Efficient NetB0*	95%
Model 4	Grade 4 Brain Tumor	*Efficient NetB7*	98%

4. Conclusion

Based on the results in this study, it can be concluded that the model scenario tested can exceed the accuracy results obtained by previous studies. Model 4 EfficientNetB7 scenario using Hyperparameter Tuning obtained 98% accuracy, superior to previous research from Agus Eko and his colleagues with 96% accuracy [1] and 3 other scenarios. The model 3 scenario, namely the EfficientNetB0 model with Hyperparameter Tuning, gets an accuracy of 95%, the model 2 scenario, namely the EfficientNetB7 model without using Hyperparameter Tuning, the accuracy value obtained is 93%, and the accuracy value obtained from the scenario 1 EfficientNetB0 model without using Hyperparameter Tuning, the accuracy value is 91%. Using the Hyperparameter Tuning technique can help improve the accuracy of the proposed model. In addition, the right augmentation technique can increase the accuracy value. Based on the results obtained from the test results in this study, it can be concluded that the proposed model is effective in classifying brain tumors.

Suggestions for further research with similar research topics, namely by considering the dataset used. In this study, there are several modifications to the preprocessing of the dataset, thus it is advisable that these stages can be used for further research to find out what preprocessing technique is better for the data so that it will get maximum performance and results.

DOI: https://doi.org/10.29207/resti.v6i4.4119

Reference

[1] A. E. Minarno, M. Hazmi Cokro Mandiri, Y. Munarko, and H. Hariyady, “Convolutional Neural Network with Hyperparameter Tuning for Brain Tumor Classification,” Kinet. Game Technol. Inf. Syst. Comput. Network, Comput. Electron. Control, vol. 4, 2021, https://doi.org/10.22219/kinetik.v6i2.1219

[2] A. Yentür, “Brain MRI Image Classification using kernel SVM,” no. June, 2021

[3] K. Pattabiraman, S. K. Muchnik, and N. Sestan, “The evolution of the human brain and disease susceptibility,” Curr. Opin. Genet. Dev., vol. 65, pp. 91–97, 2020, https://doi.org/10.1016/j.gde.2020.05.004

[4] Z. Hu, J. Tang, Z. Wang, K. Zhang, L. Zhang, and Q. Sun, “Deep learning for image-based cancer detection and diagnosis − A survey,” Pattern Recognit., vol. 83, pp. 134–149, 2018, https://doi.org/10.1016/j.patcog.2018.05.014

[5] I. B. L. M. Suta, R. S. Hartati, and Y. Divayana, “Diagnosa Tumor Otak Berdasarkan Citra MRI (Magnetic Resonance Imaging),” Maj. Ilm. Teknol. Elektro, vol. 18, no. 2, 2019, DOI: https://doi.org/10.24843/MITE.2019.v18i02.P01

[6] H. A. Khan, W. Jue, M. Mushtaq, and M. U. Mushtaq, “Brain tumor classification in MRI image using convolutional neural network,” Math. Biosci. Eng., vol. 17, no. 5, pp. 6203–6216, 2020, doi: 10.3934/MBE.2020328.

[7] W. H.-P. SeNTIK and undefined 2021, “Convolution Neural Network Arsitektur Mobilenet-V2 Untuk Mendeteksi Tumor Otak,” Ejournal.Jak-Stik.Ac.Id, vol. 5, no. 1, 2021, [Online]. Available: https://ejournal.jak-stik.ac.id/index.php/sentik/article/view/2826

[8] A. S. Febrianti, T. A. Sardjono, and A. F. Babgei, “Klasifikasi Tumor Otak pada Citra Magnetic Resonance Image dengan Menggunakan Metode Support Vector Machine,” J. Tek. ITS, vol. 9, no. 1, 2020, doi: 10.12962/j23373539.v9i1.51587.

[9] M. O. Khairandish, M. Sharma, V. Jain, J. M. Chatterjee, and N. Z. Jhanjhi, “A Hybrid CNN-SVM Threshold Segmentation Approach for Tumor Detection and Classification of MRI Brain Images,” *Irbm*, vol. 1, pp. 1–10, 2021, doi: 10.1016/j.irbm.2021.06.003.

[10] J. Amin, M. Sharif, M. Yasmin, and S. L. Fernandes, “A distinctive approach in brain tumor detection and classification using MRI,” Pattern Recognit. Lett., vol. 139, pp. 118–127, 2020, https://doi.org/10.1016/j.patrec.2017.10.036

[11] M. Mittal, L. M. Goyal, S. Kaur, I. Kaur, A. Verma, and D. Jude Hemanth, “Deep learning based enhanced tumor segmentation approach for MR brain images,” Appl. Soft Comput. J., vol. 78, pp. 346–354, 2019, https://doi.org/10.1016/j.asoc.2019.02.036

[12] N. Abiwinanda, M. Hanif, ST Hesaputra, A. Handayani, and TR Mengko, Brain Tumor Classification Using Convolutional Neural Network, vol. 68, no. 1. Springer Singapore, 2018.https://link.springer.com/chapter/10.1007/978-981-10-9035-6_33

[13] Umaei, A., Hassan, M. M., Hassan, M. R., Alelaiwi, A., & Fortino, G. (2019). ”A Hybrid Feature Extraction Method with Regularized Extreme Learning Machine for Brain Tumor Classification”. IEEE Access, 1– 1. doi:10.1109/access.2019.2904145.

[14] A. Yang, X. Yang, W. Wu, H. Liu, and Y. Zhuansun, “Research on feature extraction of tumor image based on convolutional neural network,” IEEE Access, vol. 7, pp. 24204–24213, 2019, doi: 10.1109/ACCESS.2019.2897131.

[15] D. R. Nayak, N. Padhy, P. K. Mallick, M. Zymbler, and S. Kumar, “Brain Tumor Classification Using Dense Efficient-Net,” *Axioms*, vol. 11, no. 1, 2022, doi: 10.3390/axioms11010034.

[16] S. Bhuvaji, A. Kadam, P. Bhumkar, and S. Dedge, “Brain Tumor Classification (MRI) | Kaggle, ”2020.

[17] X. Xiao, M. Yan, S. Basodi, C. Ji, and Y. Pan, "Efficient Hyperparameter Optimization in Deep Learning Using a Variable Length Genetic Algorithm," arXiv, 2020. https://doi.org/10.48550/arXiv.2006.12703

[18] MP Ranjit, G. Ganapathy, K. Sridhar, and V. Arumugham, "Efficient deep learning hyperparameter tuning using cloud infrastructure: Intelligent distributed hyperparameter tuning with Bayesian optimization in the cloud," IEEE Int. Conf. Cloud Comput. CLOUD, vol. 2019-July, pp. 520–522,2019.https://doi.org/10.1109/CLOUD.2019.00097

[19] Y Lee, SM Park, and KB Sim, "Optimal hyperparameter tuning of convolutional neural networks based on the parameter-setting-free harmony search algorithm," Optics (Stuttg)., vol. 172, no. July, pp. 359–367, 2018. https://doi.org/10.1016/j.ijleo.2018.07.044

[20] M. Koklu, I. Cinar, and Y. S. Taspinar, “Classification of rice varieties with deep learning methods,” Comput. Electron. Agric., vol. 187, no. June, p. 106285, 2021, doi: 10.1016/j.compag.2021.106285.

[21] Annisa Fitria Nurjannah, Andi Shafira Dyah Kurniasari, Zamah Sari, and Yufis Azhar, “Pneumonia Image Classification Using CNN with Max Pooling and Average Pooling,” J. RESTI (Rekayasa Sist. dan Teknol. Informasi), vol. 6, no. 2, pp. 330–338, 2022, doi: 10.29207/resti.v6i2.4001.

[22] Ulfah Nur Oktaviana and Yufis Azhar, “Klasifikasi Sampah Menggunakan Ensemble DenseNet169,” *Resti*, vol. 1, no. 1, pp. 19–25, 2021, doi**:** https://doi.org/10.29207/resti.v5i6.3673

DOI: https://doi.org/10.29207/resti.v6i4.4119