Medical Hypotheses 134 (2020) 109531

Contents lists available at ScienceDirect

Medical Hypotheses

journal homepage: www.elsevier.com/locate/mehy

BrainMRNet: Brain tumor detection using magnetic resonance images with a novel convolutional neural network model

Mesut Toğaçar[a,*], Burhan Ergen[b], Zafer Cömert[c]

[a] Department of Computer Technology, Fırat University, Elazig, Turkey
[b] Department of Computer Engineering, Faculty of Engineering, Fırat University, Elazig, Turkey
[c] Department of Software Engineering, Faculty of Engineering, Samsun University, Samsun, Turkey

ARTICLE INFO

Keywords:
Biomedical signal processing
Attention module
Magnetic resonance image
Hypercolumn technique
Brain tumor

ABSTRACT

A brain tumor is a mass that grows unevenly in the brain and directly affects human life. This mass occurs spontaneously because of the tissues surrounding the brain or the skull. Surgical methods are generally preferred for the treatment of the brain tumor. Recently, models of deep learning in the diagnosis and treatment of diseases in the biomedical field have gained intense interest. In this study, we propose a new convolutional neural network model named BrainMRNet. This architecture is built on attention modules and hypercolumn technique; it has a residual network. Firstly, image is preprocessed in BrainMRNet. Then, this step is transferred to attention modules using image augmentation techniques for each image. Attention modules select important areas of the image and the image is transferred to convolutional layers. One of the most important techniques that the BrainMRNet model uses in the convolutional layers is hypercolumn. With the help of this technique, the features extracted from each layer of the BrainMRNet model are retained by the array structure in the last layer. The aim is to select the best and the most efficient features among the features maintained in the array. Accessible magnetic resonance images were used to detect brain tumor with the BrainMRNet model. BrainMRNet model is more successful than the pre-trained convolutional neural network models (AlexNet, GoogleNet, VGG-16) used in this study. The classification success achieved with the BrainMRNet model was 96.05%.

Introduction

A brain tumor is a mass that is formed inside the brain by the tissues surrounding the brain or the skull and directly affects human life. This mass is divided into two parts as benign or malignant. These tumors grow unevenly in the brain and apply pressure around them [1]. With the effect of pressure, it causes various disorders in the brain that affects the body. Symptoms of such disorders in humans; dizziness, headache, fainting attacks, paralysis and so on. Unlike benign tumors, malignant tumors grow unevenly, damaging the surrounding tissues. In the treatment of brain tumor, surgical methods are generally preferred [2]. However, if the tumor to be taken by surgery is in a vital area, different treatment methods such as medication, radiation, etc. are applied [3]. In 2019, it is estimated that there are approximately 700 thousand people living with brain tumors in America. Approximately 86 thousand of these cases were diagnosed in 2019 [4]. Among these patients, 60.800 were found to be benign and 26.170 were found to be malignant. The survival rate of malignant patients in the US is 35% [5].

Recently, deep learning models has announced its name in the field of biomedical applications. The network of deep learning consists of numerous hidden layers. In addition, this model performs the learning process on the dataset automatically [6,7]. Convolutional neural network (CNN) architecture is a deep learning model. In this study, we proposed a new model that performs classification on brain MR images. The proposed model consists of attention modules, hypercolumn technique and residual blocks. The reason for using attention modules in the proposed model is to focus on the diseased area [8]. In addition, each activation feature extracted in the network section of the BrainMRNet model by the hypercolumn technique is kept in a series of structures in the last layer. The aim here is to select the most efficient ones among the features kept in the array structure [9].

Many techniques and methods have been proposed for the classification of brain tumors. Muhammad Sajjad *et al.* [10] proposed a novel CNN model for the classification of brain tumor. They chose the tumor site by segmentation before MR images. In the next step, they augmented the dataset. Then, they performed the classification process with the proposed CNN. Their classification success rate was 94.58%. Shiu Kumar *et al.* [11] used long short-term memory (LSTM) network

* Corresponding author.
E-mail addresses: mtogacar@firat.edu.tr (M. Toğaçar), bergen@firat.edu.tr (B. Ergen), zcomert@samsun.edu.tr (Z. Cömert).

https://doi.org/10.1016/j.mehy.2019.109531
Received 28 November 2019; Received in revised form 11 December 2019; Accepted 14 December 2019

and machine learning methods in the classification of brain tumors. They performed segmentation and then augmentation of the dataset. They conducted the training of the dataset with the LSTM model. They performed the classification process with support vector machines (SVM). The best classification success in their study was 78.33%. Varuna Shree *et al.* [12] used brain MR images in their studies. The data set they use is divided into two classes (normal and abnormal). They first applied the gray level co-occurrence matrix (GLCM) method to the dataset. They used the GLCM method together with a probabilistic neural network (PNN) model. PNN is a Bayesian algorithm based feedforward model. The success rate was 95%. Muhammad Nazir *et al.* [13] divided the brain MR images into two classes: benign and malignant. As a preprocessing step of the dataset, they applied filter methods to eliminate noise on the images. They then extracted features over the average color moment of each image. They classified the extracted features by artificial neural network (ANN). In their study, the classification success rate was 91.8%. Kanmani *et al.* [14] divided the MR dataset into two classes as normal and abnormal. They used the threshold based region optimization (TBRO) method to increase the efficiency of classification accuracy. They performed segmentation with this method. The experimental success rate of the proposed technique was 96.57%. Heba Ahmad *et al.* [15] divided the MR dataset into two classes, normal and abnormal. They proposed a combined ANN model and a gray wolf optimizer (GWO) optimization method. The success rate of GWO-ANN was 98.91%.

The sections of this article are organized as follows: Information on material and methods is given in Section 2. The proposed model explained detailed in Section 3. The analysis results of the experiment are given in Section 4. In Sections 5 and 6, discussion and concluding remarks are given, respectively.

Material and models

Dataset

Dataset used in this study consists of free accessible MR images categorized into two classes as normal and tumor [16]. The images in the dataset were collected by field experts, such as doctors and radiologists and shared on the internet. The total number of images are 253 and each image was obtained from the volunteer patients. Therefore, the dataset has a heterogeneous structure. The number of tumor images is 155 while the number of normal samples is 98. The resolution of the images is not stable and the image quality is not high. The images have been converted to JPEG format. Sample images of the classes in the dataset are shown in Fig. 1.

CNN models

In this section, AlexNet, GoogleNet and VGG-16 architectures are mentioned. The overall structure of the CNN architecture includes convolutional layers, fully connected layers and pooling layers, etc. [17,18]. The convolutional layer circulates filters (3x3, 5x5, etc.) on each input image and extracts the local discriminative features. It then transfers the extracted features to the next layer [19]. In this process, the features are kept in the activation maps. The purpose of activation maps is to select the most efficient features. The pooling layer reduces the image size and speeds up the process. This case reduces the cost of architecture [20]. The fully connected layers select efficient features according to probabilities and transfer them to the layer on which the classification process will be performed along with class possibilities [21]. The schematic design of the CNN architectures is shown in Fig. 2.

In this study, CNN architectures classified the dataset by transfer learning method. The Softmax function as a classification layer was used in these architectures. In the architecture of CNNs, generally preferred parameters and values have been used. The size of the input image of the AlexNet architecture is $227x227$ pixels. The input size of other CNN architectures was $224x224$ pixels. In addition, the features obtained from the last layers of each CNN giving 1000 features were classified with the Softmax function. The values of the parameters used in the CNNs are given in Table 1. The mini-batch value of CNN architectures was selected as 32 during the training process. The mini-batch value is determined by the capacity of the data set and the cost of the hardware used. This value is generally preferred as a number that can be divided by the size of the dataset. The mini-batch makes it easier for them to learn by balancing convergence rates with the correct prediction of the model [22–24].

Optimization and Machine Learning Methods

SGD with Warm Restarts (SGDR) is an optimization method that gradually reduces the learning rate (LR) in the cycle used in the training phase. This method reduces the learning rate in the form of a half-cosinus curve. The SGDR method is therefore also known as a cosinus annealing method. The parameter $È_t$ represents the learning rate in the process t. $È^i_{max}$ and $È^i_{min}$ parameters determine the upper and lower limits of the desired learning rate range. $T_{current}$ keeps the number of epoch that have passed since the restart process and the parameter T_i gives the number of epochs in the update [25]. The general formula of the SGDR method is shown in Eq. (1). Parameter information of this method is as follows;

- Minimum and Maximum Learning Ratio: Specifies the range of upper and lower values of the learning ratio in the SGDR method
- Steps per epoch: Returns the value calculated by dividing the epoch-size by the mini-batch size.
- LR Decay: Decreases the upper limit (Max. LR) of the learning rate after each epoch. In this study, this value was taken as 0.8. So the maximum LR at the end of each epoch was reduced by 20%.
- Cycle Length: Represents the number of the first epoch in the cycle.
- Mult. Factor: Specifies the scale values of the epoch to restart after each cycle is completed [26].

$$È_t = È^i_{min} + \frac{1}{2}(È^i_{max} - È^i_{min})\left(1 + \cos\left(\frac{T_{current}}{T_i}\right)\prod\right) \quad (1)$$

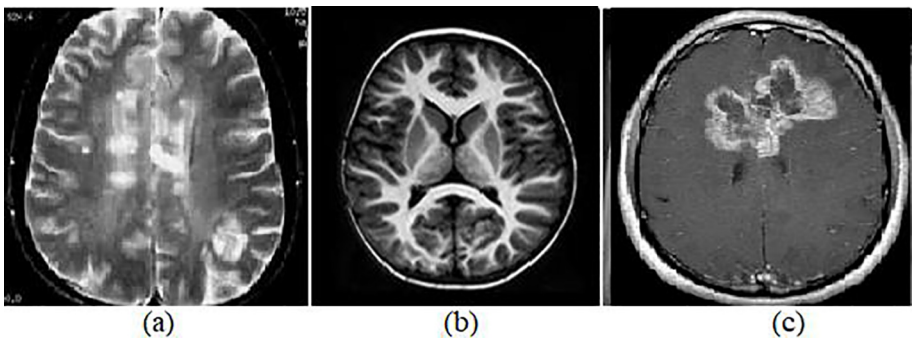

Fig. 1. (a) and (b) normal samples, (c) a tumor sample.

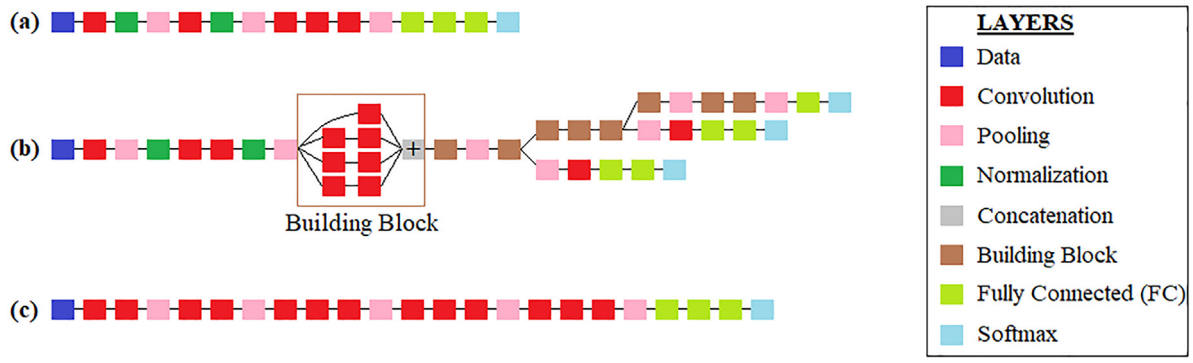

Fig. 2. The schematic representations of CNN models: (a) AlexNet, (b) GoogleNet and (c) VGG-16.

Table 1
Parameters and values of CNN architectures used in this study.

Simulation environment	CNN Architecture	Input Size	Optimization	Momentum	Decay	Mini Batch	Learning Rate
MATLAB	AlexNet	227 × 227	SGD	0.9	1e-6	32	0.0001
	GoogleNet	224 × 224					
	VGG-16	224 × 224					

The ADAM function is an effective optimization method that updates the weight parameters and learning coefficients in each batch. During the training of the model, the ADAM method helps to adjust the learning rates of the weight parameters by estimating the gradient values. ADAM uses the exponential moving averages calculated in gradient descents in its algorithm. The past gradient (m_t) and past square gradient (V_t) averages are calculated according to Eqs. (2 and 3). The variable β is used to calculate hyper parameter values. This value is between $0.9 < \beta < 0.999$ [27].

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \tag{2}$$

$$V_t = \beta_2 V_{t-1} + (1 - \beta_2) g_t^2 \tag{3}$$

The Softmax is generally preferred for classification and regression in the final layer of CNNs. The purpose of this function is to normalize nonlinear features. In the next step, the normalized features are converted to probability values and classification is performed [28,29].

A multi-layer sensor (MLP) is an advanced class neural network. The network structure of the MLP method consists of a minimum of three layers. These are input layer, hidden layer and output layer, respectively. MLP method can make weight updates in the network with backpropagation approach. Therefore, it is a controlled model. Multiple layers and non-linear activation separate MLP from a linear sensor and MLP separates non-linearly separable data [30].

Proposed CNN Model: BrainMRNet

The BrainMRNet model incorporates some of the layers of CNN architectures in its overall structure (Convolution, pooling, etc.). In addition, the dense layer is used before the classification stage. The dense layer acts as a kind of hidden layer. The dense layer carries the values of a matrix vector and is continuously updated during the back propagation. With the dense layer, the matrix size where the values are kept is changed [31]. In the BrainMRNet model, the dense layer is connected to the Softmax function, which produces 0–1 probabilities for each artificial neuron. As a result, the sum of the possibilities generated for the Softmax function is equal to one [32].

The BrainMRNet architecture is an end-to-end model. Thus, the proposed model performs images as input and distributes the probability distribution between non-encoder and decoder type classes. The proposed approach generally consists of four parts. These; convolution block attention module (CBAM) is the application of dense blocks, residual blocks and hypercolumn technique. CBAM allows to focus on the relevant area on the brain MR image. That is, it seeks important feature values on the image, searching for those values in each image and allows the model to draw attention to these values. This provides higher precision for the classification process and saves time. The CBAM module consists of a combination of two sub-modules (channel attention module and spatial attention module). Residual blocks, on the other hand, make the descent of the gradients that serve as optimization more smooth [33,34]. It helps to extract more efficient features from brain MR images with hypercolumn technique. This technique therefore contributes to the classification process and the performance of the model. The general structure of the BrainMRNet model and the preferred parameter values in this model are shown in Fig. 3.

Convolutional & dense blocks

These blocks consist of the layers shown in Fig. 4. By means of these blocks, it selects the corresponding features in the tensors processed as input. (For example, two-dimensional image or one-dimensional neuron output). Batch normalization is performed very well when used immediately after convolution and dense layers [35]. Convolution Blocks and Dense Blocks are included in the BrainMRNet model as shown in Fig. 4. The parameter values of the functions and functions used in this model are also shown in Fig. 4. There, Rectified Linear Unit (ReLU) is a non-linear multilayer activation function. If we accept the X value as the data input value, the value that ReLU will generate is between zero and X value. It does this according to $f(x) = max(0, X)$ equation [36,37]. A batch normalization normalizes each input during the mini-batch process. That is, it is used to accelerate the training of CNN architecture and balance activation values and improve the performance of the model [38,39].

CBAM module

CBAM is a forward-feed convolutional network module. The features extracted from the images examine with two modules. The CBAM module therefore consists of two modules [40]. If these modules are mentioned.

The channel attention module represents the channels of the activation maps extracted from the images. In fact, this module is useful for directing the neural network to important regions in feature maps. In

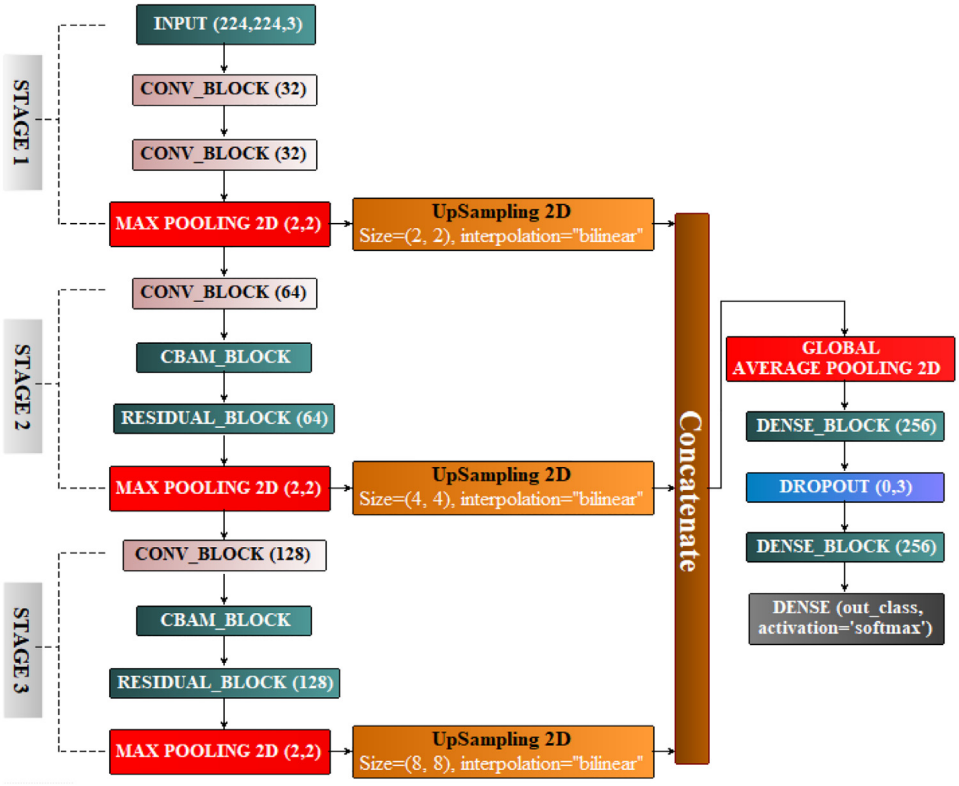

Fig. 3. Proposed model: BrainMRNet's overall design and parameters used.

addition, the channel module compresses the feature maps extracted from the images. This case allows the model to be used efficiently and uses the spatial module to obtain information about the features of the average and maximum pooling layers. The channel modules of the BrainMRNet model used two types of pool outputs. These; Maximum and average pool outputs. The spatial module uses an output pool similar to the channel module and transfers the obtained values to the convolution layer. The channel module then combines features extracted in both average and maximum pooling into a shared network. The network with the combined features consists of the MLP classifier with a hidden layer. The combined features with MLP go through the classification process. In order to calculate the spatial attention module related operations, firstly average pooling and maximum pooling operations are applied along the channel axis. Then, the two processes (average and maximum pooling) are combined to form an effective feature identifier. Finally, the obtained information is given to the after layer. [34,40]. The designs of the channel attention module and spatial attention module used in this study are shown in Fig. 5. In addition, the parameters of the functions used in the compilation part of the study are shown in Fig. 5.

As a result, CAM and SAM focus on “what” and “where respectively. These blocks contribute to performance by creating more efficient features when processing features. The general structure and parameter values of the CBAM block consisting of CAM and SAM modules are shown in Fig. 6.

Residual block

The residual blocks used in the model feed the successive layers as well as directly feed the layers beyond two or three layers. In addition, the residual block merges the layers approximately two to three jumps away by applying additional processing to the starting layer. For

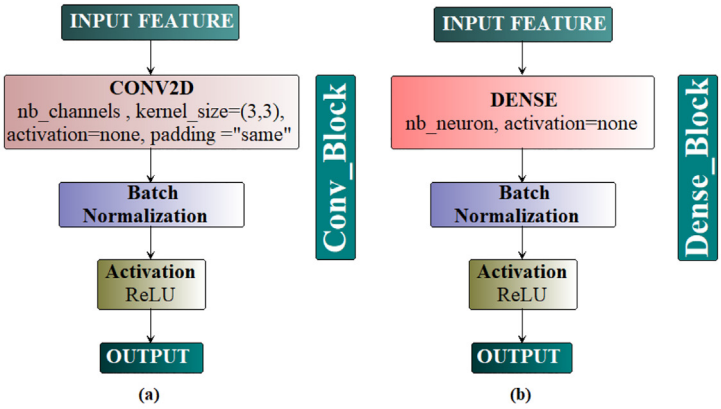

Fig. 4. General structure and parameter values of convolutional blocks and dense blocks used in the proposed model.

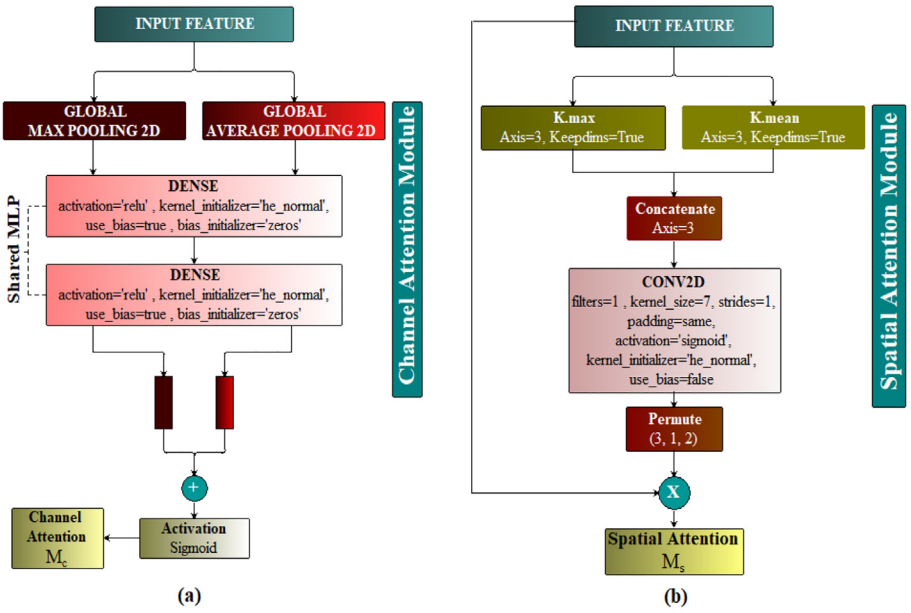

Fig. 5. Visual design of channel and spatial attention modules (CAM and SAM).

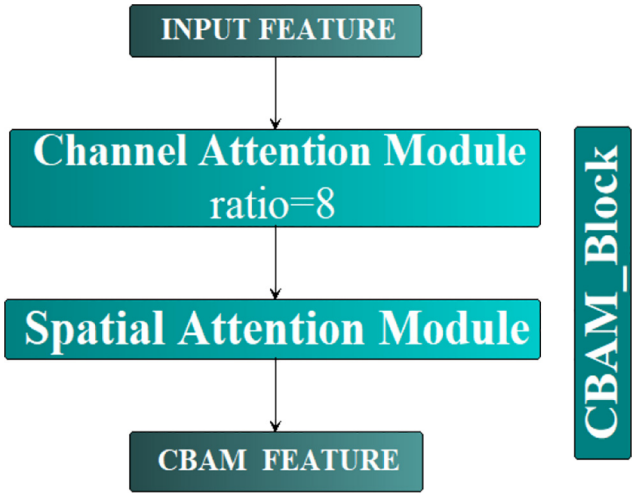

Fig. 6. General structure of CBAM block.

traditional deep learning architectures (AlexNet, VGGNets etc.), it usually performs the transfer without skipping (without shortcuts) from the convolution layers to the fully linked layers. However, when the depth of consecutive networks is increased, problems arise during the descent process of gradients and optimization methods [41,42]. To reduce this problem to the minimum level in the BrainMRNet model that we recommend, we used residual blocks in the proposed model. The operating structure of the residual blocks and the parameters used in the block structure are shown in Fig. 7.

Hypercolumn technique

The hypercolumn technique maintains the sequence of activation maps of each pixel on the image extracted along the model. And it transfers them to an array. The objective is to retrieve spatial location information (features) from the previous layers and allow for more efficient feature selection. In contrast to the hypercolumn technique, in convolutional networks, results are normally processed with the features transferred to the output of the last layer. In other words,

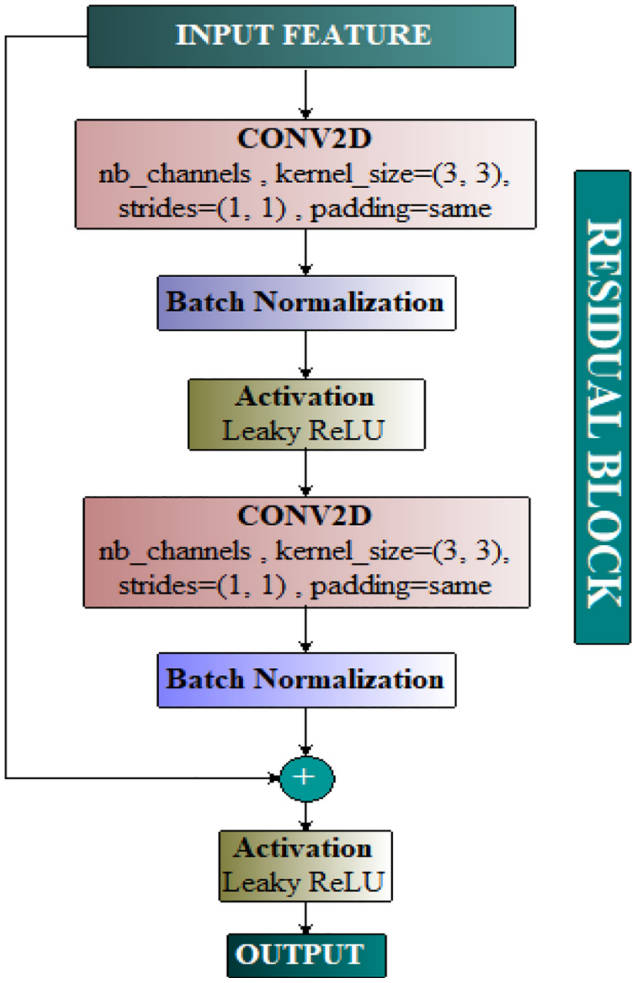

Fig. 7. Functional structure of residual blocks in the proposed model.

convolutional networks do not use the features transferred from the previous layers as a whole. The hypercolumn technique keeps the

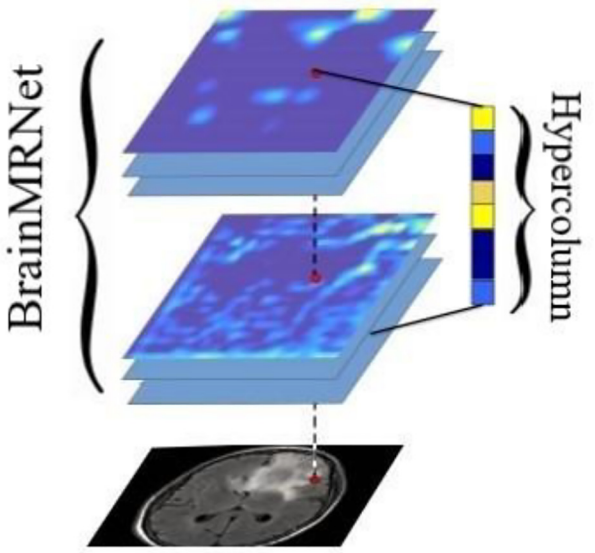

Fig. 8. The working logic of hypercolumn technique.

features of each pixel of the image processed by the model in the array structure until the model training is finished. This case increases the accuracy of the architecture. The operation of the hypercolumn technique is shown in Fig. 8. This technique involves the unknown interpolated UpSampling2D and "Concatenate" layers. The Upsampling2D function refers to the numbers of new neighboring pixels using the values of neighboring pixels around each pixel [43,44].

The general structure of the BrainMRNet architecture is shown in Fig. 3. The dropout function was used to minimize the mismatch problem during the training phase of the proposed model [45].

Experimental results

The metrics used to evaluate the analysis results of this study are; specificity, sensitivity, f-score and accuracy. These metric values were calculated by the confusion matrix. Therefore, the parameters used in the confusion matrix are; true positive (TP), true negative (TN), false positive (FP) and false negative (FN) [22,46,47].

$$\text{Se.} = \frac{TP}{TP + FN} \tag{4}$$

$$\text{Sp.} = \frac{TN}{(TN + FP)} \tag{5}$$

$$\text{Pre.} = \frac{TP}{TP + FP} \tag{6}$$

$$\text{F} - \text{Scr.} = \frac{2xTP}{2xTP + FP + FN} \tag{7}$$

$$\text{Acc.} = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

In the experimental study, CNNs (AlexNet, GoogleNet and VGG-16) were trained using transfer learning approach and were compiled on Matlab (R2019a) software. The BrainMRNet model was compiled using Python 3.6 and Python libraries. The software that compiled the codes of the proposed model was Jupyter Notebook. The software was compiled on 64-bit Windows 10 operating system. Hardware information; NVIDIA GeForce 1 GB display card, Intel © i5 – Core 2.5 GHz processor and 4 GB memory.

The values of the parameters used in the BrainMRNet architecture are shown in Table 2. In the final layer of the BrainMRNet architecture, the Softmax function was used. In addition, the input images of the proposed architecture underwent one-to-one augmentation without duplication. This case, the number of images was not increased, only the arguments for augmentation were processed. The augmentation arguments used for this particular purpose (rotation, contrast, etc.) and the hyper parameter values of these arguments are shown in Table 2. The optimization method used in the model of this study was ADAM. SGDR method was used as a parameter to facilitate learning during the training of the proposed model. In addition, the mini-batch size of the BrainMRNet architecture was set to 16. This setup was applied considering the hardware resources [48,49]. In both steps of the experiment, 70% of the dataset was reserved for training and 30% for the test data.

When Table 1 and Table 2 are examined, the mini-batch values used in the models are different. This is because present CNN models are compiled on the Matlab software. The BrainMRNet model was compiled on the Python software. Therefore, the time/memory consumption of different software on the same hardware is different.

The experiment was divided into two steps. In the first step, the brain MR images were trained with AlexNet, GoogleNet and VGG-16 architectures. In the second step, BrainMRNet was trained using the same dataset. The classification accuracy obtained from the pre-trained CNNs was 87.93% for the AlexNet, 89.66% for the GoogleNet and 84.48% for the VGG-16. In the second step, the same dataset was trained by the BrainMRNet model. The sensitivity of 96.0%, specificity of 96.08% and accuracy of 96.05% were obtained with the proposed model. The learning curves of the proposed model on the dataset are shown in Fig. 9. Experimental results were better than those of pre-trained convolutional architectures. Performance analysis of BrainMRNet architecture is shown in Table 3. In addition, the receiver operator characteristics curve & confusion matrix of the BrainMRNet model are shown in Fig. 10.

Discussion

The brain tumor is a vital disease among millions of cancer diseases. Researches have shown that the number of cases of brain tumors in the world is increasing. Since the brain is the control center of the human mechanism, damage that can occur here can directly lead to death. Therefore, early diagnosis and treatment of these diseases is extremely important. Because speed and time are important factors for the patient. Therefore, in computer-assisted devices used for early diagnosis, image processing models and methods are competing to achieve the best of these factors. The main factor that ensures this competition is human.

Table 2
Parameters used in the BrainMRNet architecture and their values.

		Optimization & Parameters	Data Augmentations Parameters	LR Scheduler & Parameters
Model	**BrainMRNet**	**ADAM**	Vertical Flip = 0.5	**SGDR**
Simulation environment	Python	$\beta_1 = 0.9$	Horizontal Flip = 0.5	Min LR = 1e-6
Input image size	224 × 224	$\beta_2 = 0.999$ Decay = 0.0	Random Brightness Contrast = 0.3	Max LR = 1e-3
Mini-batch	16		Shift Scale Rotate = 0.5	Steps Per Epoch = 5
Framework	Keras		Shift Limit = 0.2	LR Decay = 0.9Cycle Length = 10
Loss Type	Categorical Cross-entropy		Scale Limit = 0.2	Multi Factor = 2.
			Rotate Limit = 20°	

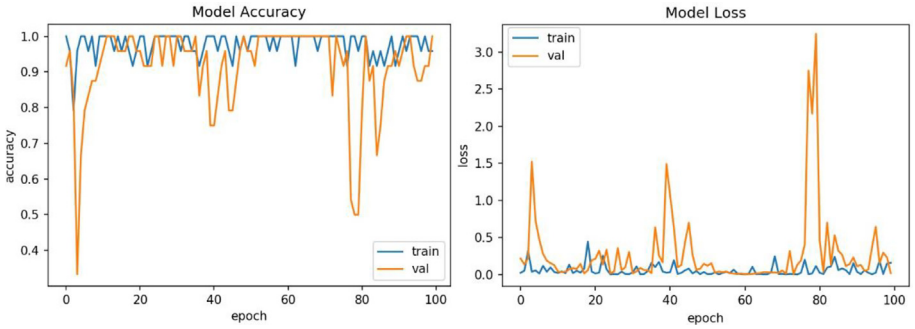

Fig. 9. The model accuracy and loss of the BrainMRNet model.

Table 3
Classification results of the experiment performed in this study.

CNN Architecture	Se. (%)	Sp. (%)	Pre. (%)	F-Scr. (%)	Acc. (%)
AlexNet	84.38	92.31	93.10	88.52	87.93
GoogleNet	84.85	96.0	96.55	90.32	89.66
VGG-16	81.25	88.46	89.66	85.25	84.48
BrainMRNet (Proposed Model)	96.0	96.08	92.31	94.12	**96.05**

The proposed model contributed to this approach. The low resolution of the dataset used prevented higher results. The comparison of the proposed model with other articles using the same dataset is shown in Table 4.

Priyansh Saxena *et al* [50] used CNN models (ResNet-50, VGG-16 and Inception-V3) in their study to train the same dataset. They carried out the training of present CNN models using transfer learning. In their study, they devoted 183 images to train. And they augmented each image at 20 times during the training. They achieved the best classification success with the ResNet-50 model. The success rate was 95%. In their study, the use of data augmentation method contributed to the classification performance. However, the BrainMRNet model has achieved better results without using the augmentation techniques and has been more successful compared to the model. In addition, Priyansh Saxena *et al.* [50] obtained the best results using the ResNet-50. The ResNet-50 has a residual block structure. It can be thought that this contributes to the success of the classification by adding depth to the training process.

Fajar Akbar *et al.* [51] performed segmentation of tumor images and obtained new images. In their work, they performed feature extraction and then classified new images with the Naive Bayes method. Their success rate was 84.17%. The method they use as segmentation process in their studies appeared to be inefficient in the analysis of their study. This case affected the results of the study. In this study, we managed to focus on the regional area within the image by using attention modules instead of segmentation method. This also contributed to the classification results.

Conclusion

With this study, a new model that can distinguish between abnormal and normal images on brain MR images has emerged. The data set used consists of two classes. These classes are; abnormal image data in which the mass was detected and normal image data. The classification accuracy of the BrainMRNet model has been superior compared to the previous studies used the same dataset. The most important distinguishing features of BrainMRNet model are given as follows:

- The proposed model focused on the relevant region on the MR images using attention modules.
- The using the hypercolumn technique, feature maps extracted at each stage in the model were transferred to the sequence structure in the last layer. Thus, all the features of a pixel from the input layer to the output layer were retained and the most efficient features were transferred to the classification process.
- By using residual blocks, the steps that negatively affect performance in the depth of the model were minimized.

The classification success achieved in this study was 96.05%. In future studies, we think to make the proposed model a model that will make it possible to use it on different medical images and in different fields [52].

Funding

There is no funding source for this article.

Open source code

The source code of the BrainMRNet model written in Python

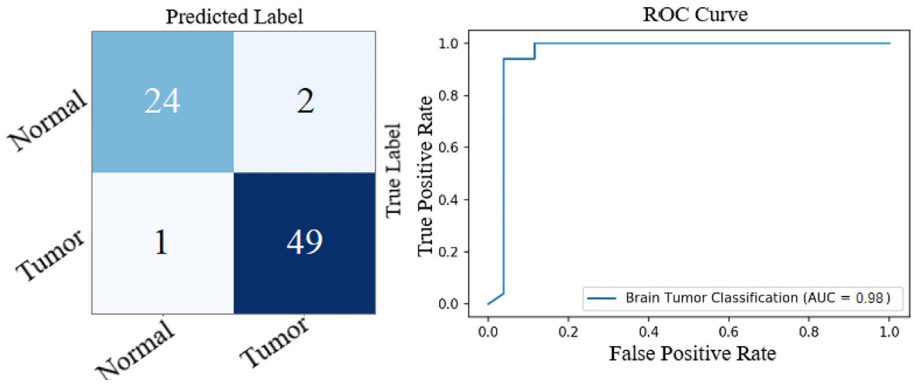

Fig. 10. The confusion matrix and ROC curve of the BrainMRNet model.

Table 4
Comparison of the analysis results of other articles using the same dataset with the analysis results of the BrainMRNet architecture.

Article	Date	Models/Methods	Classifier	Acc. (%)
Priyansh Saxena *et al.* [50]	2019	ResNet-50 /Augmentation	Softmax	95.0
Fajar Akbar *et al.* [51]	2019	The Naive Bayes	ELM	84.17
Proposed Model (BrainMRNet)	2019	Hypercolumn & Attention Module & Residual Block	Dense Layer & Softmax	**96.05**

programming language can be downloaded from https://github.com/mtogacar/BrainMRNet.

Ethical approval

This article does not contain any data, or other information from studies or experimentation, with the involvement of human or animal subjects.

Conflicts of interest

The authors declare that there is no conflict to interest related to this paper.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.mehy.2019.109531.

References

[1] Grisold W, Grisold A. Cancer around the brain. Neuro-Oncol Pract 2014;1:13–21. https://doi.org/10.1093/nop/npt002.
[2] Kheirollahi M, Dashti S, Khalaj Z, Nazemroaia F, Mahzouni P. Brain tumors: Special characters for research and banking. Adv Biomed Res 2015;4:4. https://doi.org/10.4103/2277-9175.148261.
[3] Lukoff J, Olmos J. minimizing medical radiation exposure by incorporating a new radiation “vital sign” into the electronic medical record: quality of care and patient safety. Perm J 2017;21:17. https://doi.org/10.7812/TPP/17-007.
[4] Brain Quick. Tumor facts. Cent Brain Tumor Regist United States 2019.
[5] Brain Central. Tumor registry of the United States. CBTRUS 2019.
[6] Togacar M, Ergen B, Sertkaya ME. Subclass separation of white blood cell images using convolutional neural network models. Elektron Ir Elektrotechnika 2019;25:63–8. https://doi.org/10.5755/j01.eie.25.5.24358.
[7] Budak Ü, Cömert Z, Çıbuk M, Şengür A. DCCMED-Net: Densely connected and concatenated multi Encoder-Decoder CNNs for retinal vessel extraction from fundus images. Med Hypotheses 2020;134:109426https://doi.org/10.1016/j.mehy.2019.109426.
[8] Shao F, Wang X, Meng F, Zhu J, Wang D, Dai J. Improved faster R-CNN traffic sign detection based on a second region of interest and highly possible regions proposal network. Sensors (Basel) 2019;19:2288. https://doi.org/10.3390/s19102288.
[9] Hariharan B, Arbeláez P, Girshick R, Malik J. Object instance segmentation and fine-grained localization using hypercolumns. IEEE Trans Pattern Anal Mach Intell 2017;39:627–39. https://doi.org/10.1109/TPAMI.2016.2578328.
[10] Sajjad M, Khan S, Muhammad K, Wu W, Ullah A, Baik SW. Multi-grade brain tumor classification using deep CNN with extensive data augmentation. J Comput Sci 2019;30:174–82. https://doi.org/10.1016/j.jocs.2018.12.003.
[11] Kumar S, Sharma A, Tsunoda T. Brain wave classification using long short-term memory network based OPTICAL predictor. Sci Rep 2019;9:1–13. https://doi.org/10.1038/s41598-019-45605-1.
[12] Varuna Shree N, Kumar TNR. Identification and classification of brain tumor MRI images with feature extraction using DWT and probabilistic neural network. Brain Informatics 2018;5:23–30. https://doi.org/10.1007/s40708-017-0075-5.
[13] Nazir M, Wahid F, Ali Khan S. A simple and intelligent approach for brain MRI classification. J Intell Fuzzy Syst 2015;28:1127–35. https://doi.org/10.3233/IFS-141396.
[14] Kanmani P, Marikkannu P. MRI brain images classification: A multi-level threshold based region optimization technique. J Med Syst 2018;42:1–12. https://doi.org/10.1007/s10916-018-0915-8.
[15] Ahmed HM, Youssef BAB, Elkorany AS, Saleeb AA, Abd El-Samie F. Hybrid gray wolf optimizer–artificial neural network classification approach for magnetic resonance brain images. Appl Opt 2018;57:B25–31. https://doi.org/10.1364/ao.57.000B25.
[16] Chakrabarty N. Brain MRI Images for Brain Tumor Detection | Kaggle n.d. https://www.kaggle.com/navoneel/brain-mri-images-for-brain-tumor-detection (accessed June 10, 2019).
[17] Toğaçar M, Ergen B, Cömert Z. A deep feature learning model for pneumonia detection applying a combination of mRMR feature selection and machine learning models. IRBM 2019. https://doi.org/10.1016/j.irbm.2019.10.006.
[18] Budak Ü, Cömert Z, Rashid ZN, Şengür A, Çıbuk M. Computer-aided diagnosis system combining FCN and Bi-LSTM model for efficient breast cancer detection from histopathological images. Appl Soft Comput 2019;85:105765https://doi.org/10.1016/j.asoc.2019.105765.
[19] Cömert Z, Kocamaz AF. Fetal hypoxia detection based on deep convolutional neural network with transfer learning approach BT - software engineering and algorithms in intelligent systems. Cham: Springer International Publishing; 2019. p. 239–48.
[20] Chen B, Liu J, Sun J, Liu J. Flowers Classification via Deep Learning Models. 2019.
[21] Huang J, Dwivedi K, Roig G. Deep Anchored Convolutional Neural Networks 2019.
[22] Toğaçar M, Ergen B, Sertkaya ME. Zatürre Hastalığının Derin Öğrenme Modeli ile Tespiti Detection of Pneumonia with Deep Learning Model 2019;31:223–30.
[23] Toğaçar M, Ergen B. Deep Learning Approach for Classification of Breast Cancer. 2018 Int. Conf. Artif. Intell. Data Process., 2018, p. 1–5. doi:10.1109/idap.2018.8620802.
[24] Başaran E, Cömert Z, Şengür A, Budak Ü, Çelik Y, Toğaçar M. Chronic Tympanic Membrane Diagnosis based on Deep Convolutional Neural Network. 2019 4th Int. Conf. Comput. Sci. Eng., 2019, p. 1–4. doi:10.1109/UBMK.2019.8907070.
[25] Park A, Chute C, Rajpurkar P, Lou J, Ball RL, Shpanskaya K, et al. Deep learning-assisted diagnosis of cerebral aneurysms using the HeadXNet Model. JAMA Netw Open 2019;2:e195600https://doi.org/10.1001/jamanetworkopen.2019.5600.
[26] Loshchilov I, Hutter F. SGDR: Stochastic Gradient Descent with Warm Restarts 2017:1–16.
[27] Roy SK, Mhammedi Z, Harandi M. Geometry aware constrained optimization techniques for deep learning. IEEE/CVF Conf Comput Vis Pattern Recognit 2018;2018:4460–9.
[28] Toğaçar M, Ergen B, Cömert Z. Application of breast cancer diagnosis based on a combination of convolutional neural networks, ridge regression and linear discriminant analysis using invasive breast cancer images processed with auto-encoders. Med Hypotheses 2020::109503https://doi.org/10.1016/j.mehy.2019.109503.
[29] Kanai S, Fujiwara Y, Yamanaka Y, Adachi S. Sigsoftmax: Reanalysis of the Softmax Bottleneck 2018.
[30] Salah LB, Fourati F. Deep MLP neural network control of bioreactor. 2019 10th Int. Renew. Energy Congr., 2019, p. 1–5. doi:10.1109/irec.2019.8754572.
[31] Babaeizadeh M, Smaragdis P, Campbell RH. A simple yet effective method to prune dense layers of neural networks. ICLR 2017, International Conference on Learning Representations. 2017. p. 1–10.
[32] Gao B, Pavel L. On the Properties of the Softmax Function with Application in Game Theory and On the Properties of the Softmax Function with Application in Game Theory and Reinforcement Learning 2018.
[33] Chao H, Fenhua W, Ran Z. Sign Language Recognition Based on CBAM-ResNet. Proc. 2019 Int. Conf. Artif. Intell. Adv. Manuf., New York, NY, USA: ACM; 2019, p. 48:1–48:6. doi:10.1145/3358331.3358379.
[34] Woo S, Park J, Lee J-Y, Kweon IS. CBAM: Convolutional Block Attention Module. Eccv2018 n.d.
[35] Wang S, Wang H, Xiang S, Yu L. Densely connected convolutional network block based autoencoder for panorama map compression. Signal Process Image Commun 2020;80:115678https://doi.org/10.1016/j.image.2019.115678.
[36] Eckle K, Schmidt-Hieber J. A comparison of deep networks with ReLU activation function and linear spline-type methods. Neural Networks 2019;110:232–42. https://doi.org/10.1016/j.neunet.2018.11.005.
[37] Toğaçar M, Ergen B, Cömert Z. Detection of lung cancer on chest CT images using minimum redundancy maximum relevance feature selection method with convolutional neural networks. Biocybern Biomed Eng 2019. https://doi.org/10.1016/j.bbe.2019.11.004.
[38] Li Y, Wang N, Shi J, Hou X, Liu J. Adaptive Batch Normalization for practical domain adaptation. Pattern Recognit 2018;80:109–17. https://doi.org/10.1016/j.patcog.2018.03.005.
[39] Toğaçar M, Ergen B. Biyomedikal Görüntülerde Derin Öğrenme ile Mevcut Yöntemlerin Kıyaslanması. Fırat Üniversitesi Mühendislik Bilim Derg 2019;31:109–21.
[40] Wang D, Gao F, Dong J, Wang S. Change Detection in Synthetic Aperture Radar Images based on Convolutional Block Attention Module. 2019 10th Int. Work. Anal. Multitemporal Remote Sens. Images, 2019, p. 1–4. doi:10.1109/Multi-Temp.2019.8866962.
[41] Zhang K, Sun M, Han TX, Yuan X, Guo L, Liu T. Residual networks of residual networks: multilevel residual networks. IEEE Trans Circuits Syst Video Technol 2018;28:1303–14. https://doi.org/10.1109/tcsvt.2017.2654543.
[42] Liu F, Liu J, Fu J, LU H. Improving Residual Block for Semantic Image Segmentation. 2018 IEEE Fourth Int. Conf. Multimed. Big Data, 2018, p. 1–5. doi:10.1109/Bigmm.2018.8499452.
[43] Pilly PK, Stepp ND, Liapis Y, Payton DW, Srinivasa N. Hypercolumn sparsification for low-power convolutional neural networks. J Emerg Technol Comput Syst 2019;15(20). https://doi.org/10.1145/3304104. 1–20:16.
[44] Hariharan B, Arbeláez P, Girshick R, Malik J. Hypercolumns for object segmentation and fine-grained localization. Proc IEEE Comput Soc Conf Comput Vis Pattern Recognit 2015:447–56. https://doi.org/10.1109/CVPR.2015.7298642.
[45] Ramachandram D, W Taylor G. Skin lesion segmentation using deep hypercolumn descriptors. vol. 3. 2018. doi:10.15353/vsnl.v3i1.173.
[46] Cömert Z, Şengür A, Budak Ü, Kocamaz AF. Prediction of intrapartum fetal hypoxia considering feature selection algorithms and machine learning models. Heal Inf Sci

Syst 2019;7:17. https://doi.org/10.1007/s13755-019-0079-z.
[47] Toğaçar M, Özkurt KB, Ergen B, Cömert Z. BreastNet: A novel convolutional neural network model through histopathological images for the diagnosis of breast cancer. Phys A Stat Mech Its Appl 2020.. https://doi.org/10.1016/j.physa.2019.123592.
[48] Cömert Z, Kocamaz AF. A study based on gray level co-occurrence matrix and neural network community for determination of hypoxic fetuses. Int Artif Intell Data Process Symp, TR 2016:569–73. https://doi.org/10.13140/RG.2.2.23901.00489.
[49] Yang Z, Wang C, Zhang Z, Li J. Mini-batch algorithms with online step size. Knowledge-Based Syst 2019;165:228–40. https://doi.org/10.1016/j.knosys.2018.11.031.
[50] Saxena P, Maheshwari A, Maheshwari S. Predictive modeling of brain tumor: A Deep learning approach 2019.
[51] Akbar F, Rais AN, Sobari IA, Zuama RA, Rudiarto B. Analisis performa algoritma naive bayes pada deteksi otomatis citra mri. JITK (Jurnal Ilmu Pengetah Dan Teknol Komputer) 2019;5:37–42. https://doi.org/10.33480/jitk.v5i1.586.
[52] Sertkaya ME, Ergen B, Togacar M. Diagnosis of Eye Retinal Diseases Based on Convolutional Neural Networks Using Optical Coherence Images. 2019 23rd Int. Conf. Electron., 2019, p. 1–5. doi:10.1109/electronics.2019.8765579.