Received: August 27, 2023. Revised: October 24, 2023.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

A New Voting of Convolutional Neural Networks for Brain Tumor Detection Based on MRI Images

Irwan Budi Santoso[1]* **Shoffin Nahwa Utama[1]** **Supriyono[1]**

[1]*Department of Informatics Engineering, Universitas Islam Negeri Maulana Malik Ibrahim, Malang, Indonesia*
* Corresponding author's Email: irwan@ti.uin-malang.ac.id

Abstract: Detection of brain tumors based on magnetic resonance imaging (MRI) images is essential for follow-up examinations. Several CNN models have been proposed before to get the best performance in detecting brain tumors. However, it is still necessary to improve performance due to complex brain structures, varying tumor shapes and sizes, and the position of brain tumors. Therefore, we propose a new voting of convolutional neural networks (CNNs) based on MRI images to detect brain tumors. CNN in the proposed method has three network paths, each involving a convolution process with a different kernel size. Involving different MRI image input shapes on the proposed CNN can provide different detection results, so voting is needed for the final detection. We propose a voting method with the condition that if only one proposed CNN model with a particular input shape detects a brain tumor in the MRI image, the final detection result indicates a tumor in the image. To evaluate the method's performance, we used brain MRI image datasets for tumor detection arranged into training, testing 1 (small size), and testing 2 (large size). The test results on the dataset show that the proposed method yields the best accuracy of 99.24% for testing 1 and 99.92% for testing 2. With these results, our proposed method performed better than the other methods, including VGG16, VGG19, ResNet50, MobileNetV2, InceptionV3, and Xception.

Keywords: Brain tumor, Magnetic resonance imaging, Input shape, Convolutional neural network, Voting.

1. Introduction

Brain tumors are tissues that grow due to abnormal cells in and around the brain [1]. The cause of brain tumors remains unknown. The spread of cancer cells from other parts of the body to the brain can also cause brain tumors in a person. The world health organization (WHO), in its 2018 report, released that around 9.6 million people in various places of the world died from cancer [2]. Brain tumors are generally divided into two types, namely benign and malignant tumors [3]. Benign tumors only grow in one part of a person's body and do not spread to other parts. Meanwhile, a malignant tumor (cancer) attacks the surrounding tissues and spreads to other body parts. The existence of a delay and inaccuracy in detecting a brain tumor in a person is one of the factors causing death [4, 5]. The complex structure of the human brain and the varying sizes of brain tumors pose a challenge in detecting the presence of a brain tumor. Therefore, a system or method to help detect tumors with high performance is needed in making a diagnosis.

In supporting the diagnosis, the presence of a brain tumor can be identified by radiologists using MRI. Examination with MRI is a non-invasive examination to map internal structures and specific aspects of function in the body [6]. The imaging produced by MRI can provide contrast visualization and better spatial information [7]. Detection of brain abnormalities based on these imaging results is critical in determining the presence of a tumor on an MRI image. However, in many cases, manual examination of brain tumors for these imaging results is time-consuming and error-prone. Therefore, developing a method that can automatically help diagnose these disorders is essential.

CNN is a deep learning method often used to detect a brain tumor in a person based on MRI images. Several recent studies using deep learning to detect brain tumors with MRI image input include those conducted by Kang et al.[8], Asif et al. [9], Ahmad and Choudhury [10], and Shanthi et al. [11]. These researchers used one or several CNN models to extract features of the MRI image. Furthermore, for tumor classification/detection, they used one or several machine learning (ML) classifiers with the input of this feature. Apart from ML, some used softmax layers or deep learning methods for classification. Those who used several ML classifiers obtained the final detection results by selecting the best detection results among the ML classifiers. These efforts were put to get the best tumor classification/detection results. However, due to complex brain structures, tumor shapes and sizes that vary greatly, and the position of these brain tumors [12], it is necessary to build a CNN model specifically. The CNN model has several networks involving convolution processes with different kernel sizes. In addition, the involvement of several shapes of MRI images as the input for the CNN model may give different decision results. Therefore, it is necessary to have a final detection recommendation technique, such as voting.

From the advantages and disadvantages of previous studies and potential solutions to improve tumor detection performance, we propose a new voting method of CNN for detecting brain tumors based on MRI images. The CNN model in this study was specially constructed in such a way that it could recognize the shape and size of tumors on complex brain tissue. Glioma types of brain tumors can have different shapes and sizes, almost the same as meningioma tumor types. There are times when the shape and size of the tumor is similar to healthy brain tissue. Therefore, the proposed CNN model involves a convolution process that works parallelly with different kernel sizes to extract these tumor features. This parallel convolution process is almost the same as the multi-path CNN process, which has been proven to improve classification performance for complex images [13]. In this study, we used CNN model by considering the low number of parameters and the limited training data to avoid overfitting [14]. We also used different input MRI image shapes on the proposed CNN to improve the performance. These different input shapes may provide different feature extraction results, thereby impacting performance in classification/detection [15, 16]. The consequence of using some different input shapes is that it is necessary to formulate recommendations for the final decision to detect brain tumors. In this study, we used a voting technique that, in practice, could improve classification/detection accuracy [17].

The main contributions of this study are as follows:

- The CNN model with some convolution process network paths is proposed to detect brain tumors based on MRI images. The convolution process on each network path works in parallel with different kernel sizes and is built concerning the low number of model parameters.
- The CNN model voting scheme, which involves several different sizes of input MRI images, is applied to help the proposed CNN model obtain brain tumor features in the image.
- Building a new voting algorithm for some of the proposed CNN model detection outputs to improve the performance of MRI image-based tumor detection.

In this paper, section 2 discusses the methods and results of the relevant previous research on brain tumor detection. Section 3 describes the MRI dataset for experimentation and the proposed method for brain tumor detection. Section 4 describes the experiments carried out. Section 5 contains experimental results and discussion. Finally, section 6 contains conclusions.

2. Related work

We discussed several previous studies, namely the involvement of the CNN model in detecting brain tumors, either only for feature extraction of MRI images (hybrid) or completely for feature extraction and final classification/detection.

Recent studies using CNN for feature extraction of MRI images are reported in [8, 10, 11]. Kang et al.[8] applied a hybrid scheme to the classification of brain tumors. Starting with pre-processing (cropping, resizing, and augmentation), they used thirteen CNN transfer learning models to extract MRI image features and nine ML classifiers for feature evaluation and final classification. The test results show that the combined feature extraction from DenseNet121, ResNeXt101, and MnasNet with the final classification using fully connected provided the best detection performance. A similar study was conducted by Ahmad and Choudhury [10]. After pre-processing the MRI images (active contour, thresholding, cropped, resized, RGB image), they used transfer learning of seven CNN models for feature extraction of MRI images and five ML classifiers for classification. The trial results showed that their proposed VGG19-SVM obtained the best tumor detection performance with an accuracy of 99.39%. Feature extraction of MRI images with CNN was also done by Shanthi et al. [11]. They pre-processed the MRI image with a Gaussian filter before feature extraction. At the classification stage, they used long short-term memory (LSTM) with the weights selected using the adaptive rider optimization (ARO) algorithm. Experimental results showed their proposed method achieving a maximum accuracy of 97.5%.

Previous studies using CNN for feature extraction and classification/detection of tumors based on MRI images were reported in [3, 4, 9, 18-22]. Chatterjee et al. [3] used the ResNet(2+1)D and ResNet Mixed Convolution models to classify brain tumor types. Their proposed ResNet(2+1)D combined 2D convolution followed by 1D convolution, while ResNet Mixed Convolution combined 2D and 3D convolution. The results of the trials they conducted showed that the two models were superior to the ResNet3D model.

Khan et al. [4] proposed a 23-layer CNN for brain tumor classification. They tested the model with large and small MRI image datasets. The testing with large datasets went well, but for small datasets, there was an overfitting. To overcome this problem, they combined transfer learning VGG16 with a 23-layer CNN. Their proposed model's test results obtained a classification accuracy of up to 97.8% and 100% for large and small datasets. A similar study by Asif et al. [9] used transfer learning Xception, NasNet Large, DenseNet121, and InceptionResNet-V2 and used a softmax layer at the classification stage. Their test results showed that Xception obtained the best performance with an accuracy of 99.67 % and 91.94% for large and small datasets respectively.

Younis et al. [18] used the CNN ensemble and VGG16 models to avoid overfitting. Their test results were carried out, and the ensemble of the two models resulted in accuracy in detecting brain tumors of 98.14%. Unlike what was done by Noreen et al. [19], they have combined multi-level features for early diagnosis of brain tumors. They used two deep learning models, namely Inception-v3 and DensNet201. In their first trial, they combined features extracted from the Inception-v3 module and passed them to softmax for tumor classification. Second, they combined features extracted from DensNet201 blocks and passed them to softmax for detection. The test resulted in an accuracy of 99.34 % and 99.51% for Inception-v3 and DensNet201.

Table 1. Notation list

Notation	Description
X	MRI Image Input
W	Convolution Weight
b	Convolution Bias
$f(.)$	Activation Function
Z	Convolution Output
$\tilde{Z}$	ReLU Output
$\tilde{z}$	Feature Map Part
g	Max-Pooling Output
h	Flattening Operation Output
$\tilde{h}$	Concatenation Output
y	Softmax Output
$\hat{h}$	Fully Connected Layer Output
$\tilde{y}$	Detection Result
$\hat{y}$	Voting Output
Acc	Detection Accuracy
Pre	Detection Precision
Sen	Detection Sensitivity
Spe	Detection Specificity

Rizwan et al. [20] proposed a gaussian convolutional neural network (GCNN) to detect brain tumor types. They also pre-processed some noise filters and smoothed the MRI images. The test results with Gaussian filters and GCNN obtained the best accuracy of 99.8% and 97.14% for the first and second datasets. Similar research was also conducted by Musallam et al.[21] who proposed a three-step pre-process and deep convolutional neural network (DCNN) to classify tumor types. The pre-process included cropping, denoising with a non-local mean algorithm, and histogram equalization. The test results showed that their proposed method obtained an overall classification accuracy of 98.22%. A different model for brain tumor classification was proposed by Jun et al. [22], which involved a multipath network in parallel with each path using the same size and number of convolution kernels. They used a gated channel transformation (GCT) layer to improve deep-level CNN performance. The evaluation results showed that their proposed method achieved an accuracy of 98.61% in classifying tumor types.

Combining the features of two or more CNN models is the best effort to improve classification performance and avoid overfitting. In the case of MRI image-based tumor detection with high variations in tumor size, creating a multi-path network with different kernel sizes can be a solution to obtain features from tumor images. Therefore, in this study, we propose a CNN architecture and a tumor detection scheme vs. no tumor to help the early detection. Our proposed method differs from the other methods in some aspects, they are: (i) the proposed CNN model containing several convolution process paths that work in parallel with different kernel sizes and paying attention to the low number of model parameters, (ii) the ensemble scheme of the CNN model involving some input shapes of the MRI image, (iii) using the proposed voting on some of the proposed CNN model detection outputs.

3. Material and methods

3.1 Dataset of experiment

In this study, we conducted a series of brain tumor detection trials using four brain MRI image datasets. The first dataset is brain MRI images (axial, sagittal, and coronal) for brain tumor classification downloaded from the Kaggle website [23]. The dataset consists of MRI images of tumors, including meningioma, glioma, and pituitary, and MRI images of no tumor. The dataset has been defined and segregated for training and testing. The dataset for training includes glioma, meningioma, pituitary, and no tumor, respectively 826, 822, 827, and 395. The total number of all tumor training images is 2475. The datasets for testing glioma, meningioma, pituitary, and no tumor are 100, 115, 74, and 105, respectively. The total number of tumor testing images is 289. The second dataset is MRI images of the brain for brain tumor detection, downloaded from the Kaggle website [24]. The dataset contains axial MRI images, which include 155 tumors and 98 no tumors. The third dataset is brain MRI images for detecting brain tumors, which can be retrieved from the Kaggle website [25]. The dataset includes MRI images of tumors and no tumors, totaling 1500 each. The fourth dataset is an MRI image of the brain (axial, sagittal, and coronal) obtained from the website [26]. In this study, the dataset taken from the website was only a testing dataset that included 300 gliomas, 306 meningiomas, 300 pituitaries, and 405 no tumors. Consequently, the total number of tumor images for testing is 906. An example of an MRI brain image containing tumors and no tumors is shown in Fig. 1.

From the dataset description, we arranged some datasets to evaluate our proposed method. The dataset includes datasets for training, testing 1, and testing 2, as shown in Table 2. The training dataset combines the four datasets, while the datasets for testing 1 and 2 are taken from [23] and [26].

3.2 Data pre-processing

In the training process, the input MRI image on the CNN model is required to have the same size. In the proposed scheme, some dimensions/sizes of MRI images are involved as inputs to the CNN model, namely 64x64 pixels, 128x128 pixels, 256x256 pixels, 512x512 pixels, and 224x224 pixels. Therefore, the original MRI image input, either for training or testing, is changed to that size. The next pre-process is the normalization of each pixel value of the MRI image. Normalization in this study is carried out to help process stability and convergence in the network [27, 28]. Each pixel value of the MRI image is normalized by changing the pixel value from the range [0.255] to [0.1]. This value is obtained by dividing each image pixel value by 255.

Table 2. Brain MRI images dataset for the experiment

Dataset		Training	Testing 1	Testing 2
Tumor	[23]	2475	289	-
	[24]	155	-	-
	[25]	1500	-	-
	[26]	-	-	906
	Total	**4130**	**289**	**906**
No tumor	[23]	395	105	-
	[24]	98	-	-
	[25]	1500	-	-
	[26]	-	-	405
	Total	**1993**	**105**	**405**

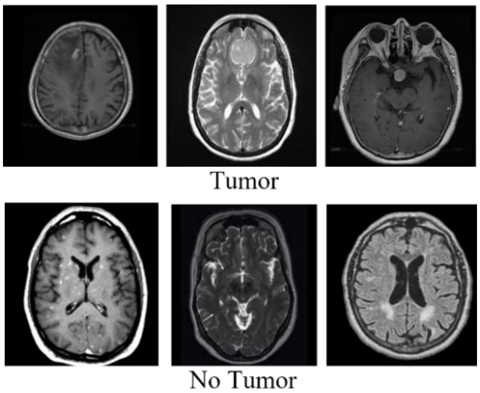

Figure. 1 The example of brain tumor MRI images and no tumor

3.3 Convolutional neural network

CNN is one of the deep learning models widely applied to image objects and has been proven to have high performance [29]. The CNN architecture proposed in this study has three network paths, each with several layers: input, convolutional, activation, and pooling layer. After the network path, there are flatten and concatenation, fully connected, and output layers. We have also prepared a notation list to help understand the notations in the proposed architecture, as shown in Table 1. Henceforth, we name the proposed CNN architecture as tpCNN. The CNN architecture is explicitly built for brain tumor detection vs. no tumor based on the MRI image, as shown in Fig. 2. In this study, the goal of making three network paths on tpCNN is to get the best brain tumor features.

3.3.1. Input layer

The input layer in this study is used to input brain MRI images into the network after the pre-processing to the convolution stage. The input MRI image on the tpCNN architecture is an RGB image with three channels with width=a and height=b. The size of a and b depends on the input shape, which is formed from the original image in the proposed scheme.

3.3.2. Convolutional layer

The convolution process in this layer is carried out for each brain MRI image included in the previous layer by shifting the filter. This process yields many MRI image feature maps to get the characteristics of the image [30, 31]. In this study, each network path on tpCNN has the convolution process, which works in parallel between one network path and another. The convolution process on each network path has the same number of filters but a different kernel size. The convolution process in the tpCNN model can be written as follows:

$$Z_i = f(W_i X + b_i), \quad i = 1, 2, 3 \tag{1}$$

with Z_i is the result of the convolution on the i th network path of the tpCNN model, W_i is the weight of the convolution on the ith network path, and b_i is the bias of the convolution on the ith network path. These weights will be updated to improve detection results in the training process [32]. In this study, the number of filters for each network path of the tpCNN model is the same. Each network path on the tpCNN architecture has three convolution layers, with 32, 16, and 8 filters for each layer. The filter size for each convolution process on one network path is the same. In the first network path, the filter size is 3x3, the second network path is 5x5, and the third network path is 7x7, as shown in Fig. 2. While the stride used in the convolution process is 1 with no padding [33].

3.3.3. Activation layer

This layer helps increase the nonlinear nature of the decision function by applying an unsaturated activation function. This study uses the rectified linear unit (ReLU) as the activation function in each convolution process [30], which is shown in the following equation:

$$\tilde{Z}_i(Z_i) = \begin{cases} Z_i, Z_i \geq 0 \\ 0, Z_i < 0 \end{cases}, \quad i = 1, 2, 3 \tag{2}$$

where $\tilde{Z}_i$ is the result of the ReLU process for the ith network path in the tpCNN model.

3.3.4. Pooling layer

The process in the pooling layer aims to reduce the size of the spatial representation of the convolution results, reduce computation, and avoid overfitting. The pooling used in this study is max-pooling [34,35], with the pooling size on each network path of the tpCNN model being 2×2. The max-pooling of the results of the convolution MRI images is written as in the following equation:

$$g_i(\tilde{\mathbf{z}}_i) = max\{\tilde{\mathbf{z}}_{ij}\}_j^4, \quad i = 1, 2, 3 \tag{3}$$

where $\tilde{\mathbf{z}}_i$ is part of the feature map in the pooling region for processes on the ith network path, and each network path has four max-pooling processes, as shown in Fig. 2.

3.3.5. Flatten and concatenation layer

In this layer, the feature matrix resulting from the last max-pooling process on each network path is reshaped in vector form with a flattening operation [36]. The flattening operation on the ith network path of the tpCNN model is written as follows:

$$h_i = flatten(g_i), \quad i = 1, 2, 3 \tag{4}$$

Flatten results are for each path of the tpCNN model before entering the fully connected layer. A concatenation process is carried out, which combines the flattened results for each path [37]. In this study, the process of combining flattened results in the tpCNN model is as follows:

$$\tilde{h} = concatenate(h_1, h_2, h_3) \tag{5}$$

where h_1, h_2, dan h_3 are the flatten results for the 1st network, 2nd network, and 3rd network paths.

3.3.6. Fully connected layer

In this layer, the weights and biases are updated against the previous layer through feedback to reduce the feature information loss. The feature vector resulting from the concatenation of flattened features on each path is connected to the output layer with a dropout process of 50% to prevent overfitting [38].

3.3.7. Output (classification) layer

The results of this fully connected layer are forwarded to the output layer to show the results of the classification and loss function. In this study, binary cross-entropy is the loss function used in the tpCNN model [38]. As for the activation function at this layer, we use softmax for binary classification (no tumor, tumor) [39]. Mathematically, the softmax function of the tpCNN model is written as in the following equation:

$$y_k(\hat{h}) = \frac{\exp(\hat{h}_k)}{\sum_{j=1}^{2} \exp(\hat{h}_j)}, k = 1, 2 \tag{6}$$

where y_k is the softmax output of the k^{th} class, and $\hat{h}$ is the fully connected layer process output.

3.4 Voting convolutional neural networks

The voting method in this study is an ensemble method similar to the bagging method [40]. In this study, no bootstrapping was performed on the data, but instead, resizing the original brain MRI image into several input shapes to the tpCNN model. The size of the brain MRI image after resizing is 64x64, 128x128, 256x256, 512x512, and 224x224, with each containing three channels (RGB). From each MRI image with a different input shape, training is carried out with the tpCNN model. The voting method is used for final detection based on the detection results by the tpCNN model. This study proposes a new voting method to improve the detection performance of the most commonly used voting, namely majority voting. Fig. 3. shows the proposed scheme using the tpCNN model and the proposed voting method.

Based on the results of the output layer of the tpCNN model, the results of brain tumor detection can be determined using the following equation:

$$\tilde{y}_r = \underset{k}{argmax}(y_{rk}), \ldots \qquad \tilde{y}_r \in \{0,1\}; r = 1, \ldots, n; k = 1, 2 \tag{7}$$

with n is the number of brain MRI image input shapes, and $\tilde{y}_r$ is the detection result (0='no tumor' and 1='tumor') in the r input shape. Detection results on each MRI image with a different input shape on tpCNN as an initial detection. The detection results are then used as input to the final detection stage using the proposed voting method, which is written as follows:

$$\hat{y} = voting(\tilde{y}_1, \tilde{y}_2, \ldots, \tilde{y}_n), \ \hat{y} \in \{0,1\} \tag{8}$$

voting on Eq. (8) is a function for final detection as a voting method proposed by following the Algorithm 1.

Algorithm 1: *voting*

```
Input: $\tilde{y}_1, \tilde{y}_2, \ldots, \tilde{y}_n$ (detection result using tpCNN)
Output: $\hat{y}$ (final detection)
$t \leftarrow 0$
for $i$= 1 to $n$ do
    $t \leftarrow t + \tilde{y}_i$
end
if $t \geq 1$ then
    $\hat{y} \leftarrow 1$
else
    $\hat{y} \leftarrow 0$
return $\hat{y}$
```

The input algorithm is $\tilde{y}_1, \tilde{y}_2, \ldots, \tilde{y}_n$ representing the results of tumor detection with tpCNN with different input shapes. t stated that many tpCNN models with varying input shapes of MRI images successfully detected brain tumors in these images. If there is a tpCNN model that successfully detects the presence of a tumor in the MRI image ($t \geq 1$), then the final detection result indicates that there is a tumor in the image ($\hat{y} \leftarrow 1$). On the other hand, if no tpCNN model can detect tumors in the MRI image, the final detection will be stated as no tumor ($\hat{y} \leftarrow 0$).

3.5 Performance evaluation

In this study, we used some measurement indicators to evaluate the results of tumor detection. These indicators include accuracy (*Acc*), precision (*Pre*), sensitivity (*Sen*), specificity (*Spe*), and *F*-score, which is determined based on true positive (*tp*),false negative (*fn*), true negative (*tn*), and false positive (*fp*) [41]. *tp* is the number of times an MRI image with no tumor is labeled as no tumor based on the detection results. *fn* is the number of times an MRI image with no tumor is labeled a tumor. *tn* is the number of times a tumor MRI image is labeled as a tumor. *fp* is the number of times a tumor MRI image is labeled as no tumor in the same way. *Acc*, *Pre*, *Sen*, *Spe*, and *F*-score values are defined in Eq. (9-13).

$$Acc = (tp + tn)/(tp + fn + tn + fp) \tag{9}$$

$$Pre = tp/(tp + fp) \tag{10}$$

$$Sen = tp/(tp + fn) \tag{11}$$

$$Spe = tn/(tn + fp) \tag{12}$$

$$F\text{-score} = 2\,(Sen)(Pre)/(Sen + Pre) \tag{13}$$

4. Experiments

We use a training and testing dataset to evaluate the proposed method for detecting brain tumors, as shown in Table 2. Meanwhile, the training and testing process uses Eq. (1-8). In this study, the success in

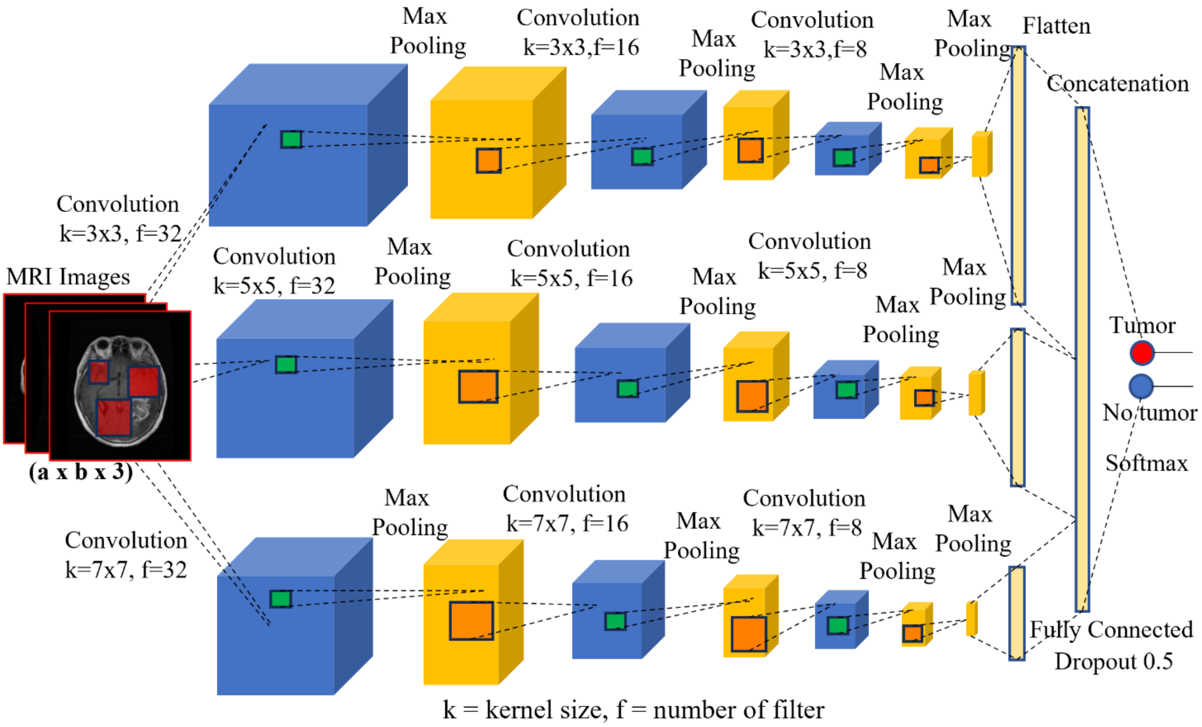

Figure. 2 Proposed CNN architecture with triple path network for tumor detection (tpCNN)

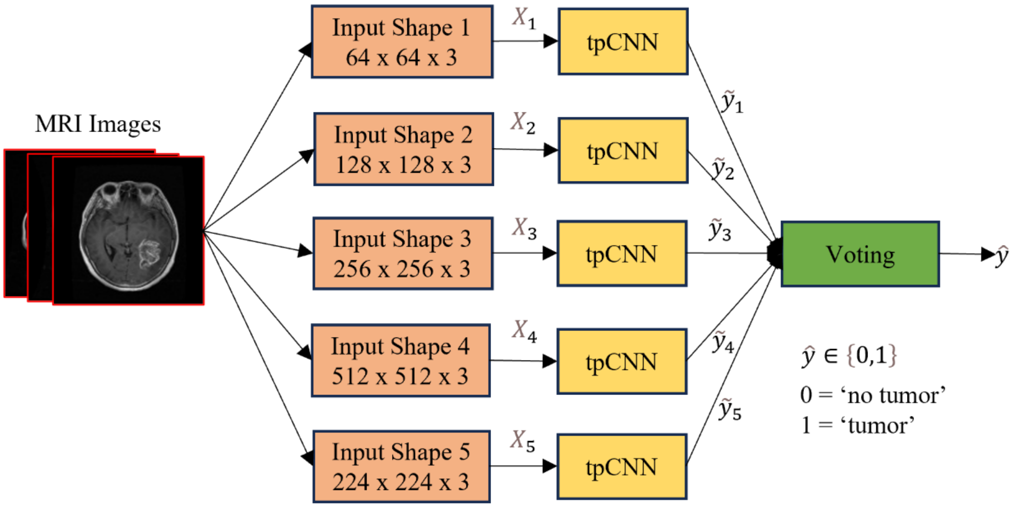

Figure. 3 The proposed scheme: voting of tpCNN with different input shapes of brain MRI images for tumor detection

detecting brain tumors on MRI images is more important as an initial step in the early disease detection. Therefore, there are more MRI images of brain tumors for training and testing than of no tumors. Based on these considerations, the evaluation of the proposed method is determined by using Eq. (9-13). In this study, we used Google Colab to implement all processes in each test scenario.

4.1 Data preparation

The number of MRI image datasets for brain tumors and no tumors for training in Table 2 is 4130 and 1993, thus, the total is 6123. The total number of MRI image datasets for brain tumors and no tumors in testing 1 is 289 and 105, so the total is 394. As for dataset testing 2, the number of brain tumor MRI image datasets and no tumors is 906 and 405, thus, the total is 1311. For the training process of the proposed tpCNN model, the training dataset with a total of 6123 is further divided into training and validation data with a composition of 90% and 10%. Accordingly, the total dataset for the training model is 5510, and the validation is 613, taken proportionally according to the number of MRI images of tumors and no tumors.

Received: August 27, 2023. Revised: October 24, 2023.

Table 3. The number of proposed model parameters

Model	Input shape	# Parameter
(1) tpCNN	64x64x3	61,482
(2) tpCNN	128x128x3	63,018
(3) tpCNN	256x256x3	70,698
(4) tpCNN	512x512x3	104,490
(5) tpCNN	224x224x3	68,202
Voting (2,3,4)	-	238,206*
Voting (1,4,5)	-	234,174*
Voting (2,4,5)	-	235,710*
Voting (1,2,3,4)	-	299,688*
Voting (1,2,3,4,5)	-	367,890*

* The total number of tpCNN parameters involved in voting with different input shapes

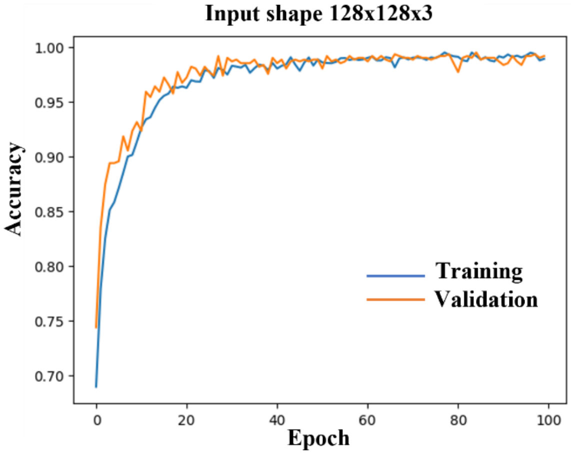

Figure. 4 The example of training and validation accuracy by tpCNN with a learning rate of 0.001

4.2 Parameters settings

Based on the scheme proposed in Fig. 3, the training of the tpCNN model is carried out with different input shapes of MRI images. The input shape of the MRI image includes input shape 1: 64 x 64 x 3, input shape 2: 128 x 128 x 3, input shape 3: 256 x 256 x 3, input shape 4: 512 x 512 x 3, and input shape 5: 224 x 224 x 3. Different input shapes cause the number of parameters of the tpCNN model for training as well to be different, as shown in Table 3.

All the tpCNN model training processes with different input shapes use the Adam optimizer because they are relatively consistent [42]. In comparison, the learning rates for all training with the proposed tpCNN model are 0.01, 0.001, 0.0001, and 0.00001. In this study, the learning rate treatment is carried out to determine the consistency of the tpCNN model for the resulting performance. At the same time, the batch sizes and epochs used are 16 and 100. Epoch 100 was chosen because the training results at that epoch were relatively stable, as reported in [14].

After training with the proposed tpCNN model, the next stage is final detection using the proposed voting method with the steps shown in the *voting* algorithm. In the proposed voting, a tumor is detected if at least one tpCNN model with a different input shape detects a tumor ($t \geq 1$). In this study, the proposed voting method was also compared with voting with $t \geq n/2$ conditions. $t \geq n/2$ means that if at least half of tpCNN models with different input shapes detect a tumor, the final detection result is a tumor. In this study, there are three voting scenarios, namely (1) involving three tpCNN models with different input shapes, (2) involving four tpCNN models with different input shapes, (3) and involving five models with different input shapes.

5. Results and discussion

In this section, we report the experimental results of our proposed method, namely the training and testing of the tpCNN model and the testing of the proposed voting method.

5.1 Experimental results

In this section, the experimental results reported are the performance of the proposed CNN method (tpCNN) on different input MRI image shapes and voting on the results of tumor detection with tpCNN. The proposed methods have been tested in each scenario and are shown in Table 4, Table 5, Table 6, and Table 7.

The results of training using the MRI image dataset in Table 2 with a predetermined validation composition, the proposed tpCNN with different input shapes (input shape 1 – input shape 5) is a non-overfitting model. Fig. 4 is an example of the tpCNN training process with an input shape of 128x128x3. At the 40th epoch, training and validation accuracy began to stabilize at the 100th epoch. The training and validation accuracy values for the epoch are almost the same, consequently there is no indication of overfitting of the tpCNN model with the input shape.

The test results on the dataset of testing 1, for tpCNN with input shape 1, obtained tumor detection accuracy of 97.72% at a learning rate of 0.0001. This result was better than tpCNN at other learning rates. In input shape 2, input shape 3, and input shape 4, the proposed tpCNN yielded the best detection accuracy at learning rates of 0.00001, 0.01, and 0.01 of 96.70%, 95.43%, and 96.45%, respectively. In input shape 5, tpCNN obtained the best accuracy at a learning rate of 0.0001 of 96.45%. Of all the input shapes tested, tpCNN on input shape 1 with a learning rate of 0.0001gave the best results.

Based on the scheme proposed in this study, detection with tpCNN is the initial detection, which

Table 4. Accuracy ($Acc(\%)$) of brain tumor detection by voting on all scenarios.

Model		Learning rate (Testing 1)				Learning rate (Testing 2)			
		0.01	0.001	0.0001	0.00001	0.01	0.001	0.0001	0.00001
tpCNN (Input Shape)	(1) 64x64x3	95.18	97.21	97.72	97.21	99.31	99.16	99.62	99.39
	(2) 128x128x3	95.69	95.18	93.65	96.70	99.16	98.86	98.09	99.85
	(3) 256x256x3	95.43	94.16	93.91	94.92	98.40	99.24	98.32	99.39
	(4) 512x512x3	96.45	94.92	96.19	94.42	98.47	98.17	99.08	98.09
	(5) 224x224x3	93.65	95.94	96.45	94.92	98.86	99.54	99.62	99.54
Voting ($t \geq n/2$)	(1,2,3)	96.70	94.92	96.19	96.45	99.24	99.62	99.31	99.85
	(1,2,4)	96.95	96.70	96.45	96.95	99.54	99.62	99.62	99.77
	(1,3,4)	96.45	95.69	96.70	96.70	99.08	99.46	99.69	99.62
	(2,3,4)	96.19	94.42	95.43	96.19	99.08	99.31	99.24	99.69
	(1,2,5)	95.94	96.70	96.19	96.45	99.62	99.69	99.77	99.85
	(1,3,5)	95.94	95.43	96.45	95.94	99.39	99.77	99.62	99.62
	(1,4,5)	95.94	95.94	96.70	96.70	99.47	99.62	99.77	99.54
	(2,3,5)	95.69	95.94	95.43	95.94	99.16	99.54	99.31	99.62
	(2,4,5)	95.94	95.69	96.45	96.19	99.24	99.54	99.77	99.62
	(3,4,5)	96.45	95.94	96.45	95.94	98.86	99.46	99.54	99.54
	(1,2,3,4)	97.21	97.46	96.95	97.72	99.54	99.85	99.77	**99.92**
	(1,2,3,4,5)	96.45	95.43	95.94	96.19	99.54	99.77	99.77	99.69
Voting ($t \geq 1$)	(1,2,3)	98.22	98.22	98.48	98.73	**99.77**	99.77	99.77	99.85
	(1,2,4)	**98.98**	98.98	98.73	**98.98**	99.69	99.77	99.77	99.85
	(1,3,4)	**98.98**	98.98	98.73	98.48	**99.77**	99.77	99.69	99.85
	(2,3,4)	97.97	98.22	97.97	98.48	99.62	99.85	**99.85**	**99.92**
	(1,2,5)	98.48	98.73	98.73	**98.98**	99.69	99.85	99.77	99.85
	(1,3,5)	98.48	98.98	98.73	98.48	99.77	99.85	99.77	99.85
	(1,4,5)	**98.98**	**99.24**	98.48	98.48	99.69	99.85	99.77	99.85
	(2,3,5)	97.72	97.97	97.72	97.21	**99.77**	99.85	**99.85**	99.85
	(2,4,5)	98.22	98.73	97.21	98.48	**99.77**	**99.92**	**99.85**	**99.92**
	(3,4,5)	97.46	97.46	97.46	97.72	**99.77**	**99.92**	**99.85**	99.77
	(1,2,3,4)	**98.98**	98.98	**98.98**	**98.98**	99.77	99.77	99.77	99.85
	(1,2,3,4,5)	**98.98**	**99.24**	**98.98**	**98.98**	99.77	99.85	99.77	99.85

becomes input to the final detection process using voting. In the voting scenario involving three tpCNN models, voting with the condition $t \geq n/2$ (majority voting) resulted in the highest detection accuracy of 96.95% involved models (1,2,4) at learning rates of 0.01 and 0.00001. While the lowest accuracy of 94.42% at a learning rate of 0.001 was obtained from the voting model (2,3,4). The highest accuracy for voting with the $t \geq 1$ condition proposed in this study 99.24%, with voting-involved models (1,4,5) at a learning rate of 0.001. While the lowest accuracy of 97.21% was obtained from voting models (2,4,5) and (2,3,5) at learning rates of 0.0001 and 0.00001, respectively.

In a voting scenario involving four tpCNN models (1,2,3,4), voting with the condition $t \geq n/2$ obtained the highest detection accuracy of 97.72% at a learning rate of 0.00001. As for the $t \geq 1$voting proposed, the best tumor detection accuracy was 98.98% at learning rates of 0.01, 0.0001, and 0.00001. For a voting scenario involving five tpCNN models, voting with the condition $t \geq n/2$ (majority voting) yielded the highest detection accuracy of 96.45% at a learning rate of 0.01. Whereas voting with the $t \geq 1$ proposed obtained the best accuracy at 99.24% at a learning rate of 0.001.

From the test results on the dataset of testing 1, voting with $t \geq n/2$ involving three models, four models, or five tpCNN models have not shown a significant increase in accuracy. It is demonstrated by voting with these conditions only obtaining the best detection accuracy equal to the best received by the tpCNN model. In contrast, the proposed $t \geq 1$ voting method can increase detection accuracy by 1.52% (99.24%-97.72%).

Judging from the detection sensitivity value at a learning rate of 0.001, the proposed $t \geq 1$ voting does not provide an increase in the sensitivity value and even tends to be constant, but there is a significant increase in the precision, the specificity, and the F-score in each voting scenario, especially those involving five tpCNN models. The precision, specificity, and F-score values yielded by the proposed voting were 98.11%, 99.31%, and 98.58%, respectively, as shown in Table 5. Based on the minimum-maximum dispersion values obtained at each learning rate, voting with $t \geq 1$ in each scenario shows a significant increase in accuracy, precision,

Table 5. Sensitivity, precision, specificity, and F-score of brain tumor detection by voting with a learning rate 0.001

Model		**Performance (%) (Testing 1)**				**Performance (%) (Testing 2)**			
		Sen	Pre	Spe	F-score	Sen	Pre	Spe	F-score
tpCNN (Input Shape)	(1) 64x64x3	99.05	91.23	96.54	94.98	99.51	97.82	99.01	98.65
	(2) 128x128x3	99.05	85.25	93.77	91.63	99.75	96.65	98.45	98.18
	(3) 256x256x3	99.05	82.54	92.39	90.04	99.75	97.82	99.01	98.78
	(4) 512x512x3	99.05	84.55	93.43	91.23	99.75	94.61	97.46	97.12
	(5) 224x224x3	99.05	87.39	94.81	92.86	99.75	98.77	99.45	99.26
Voting $(t \geq n/2)$	(1,2,3)	99.05	84.55	93.43	91.23	99.75	99.01	99.56	99.38
	(1,2,4)	99.05	89.66	95.85	94.12	99.75	99.01	99.56	99.38
	(1,3,4)	99.05	86.67	94.46	92.44	99.75	98.53	99.34	99.13
	(2,3,4)	99.05	83.20	92.73	90.43	99.75	98.04	99.12	98.89
	(1,2,5)	99.05	89.66	95.85	94.12	99.75	99.26	99.67	99.50
	(1,3,5)	99.05	85.95	94.12	92.04	99.75	99.50	99.78	99.63
	(1,4,5)	99.05	89.66	94.42	94.12	99.75	99.01	99.56	99.38
	(2,3,5)	99.05	87.39	94.81	92.86	99.75	98.77	99.45	99.26
	(2,4,5)	99.05	86.67	94.46	92.44	99.75	98.77	99.45	99.26
	(3,4,5)	99.05	87.39	94.81	92.86	99.75	98.53	99.34	99.13
	(1,2,3,4)	99.05	92.04	96.89	95.41	99.75	99.75	99.89	99.75
	(1,2,3,4,5)	99.05	85.95	94.12	92.04	99.75	99.50	99.78	99.63
Voting $(t \geq 1)$	(1,2,3)	99.05	94.55	97.92	96.74	99.51	99.75	99.89	99.63
	(1,2,4)	99.05	97.20	98.96	98.11	99.51	99.75	99.89	99.63
	(1,3,4)	99.05	97.20	98.96	98.11	99.51	99.75	99.89	99.63
	(2,3,4)	99.05	94.55	97.92	96.74	99.75	99.75	99.89	99.75
	(1,2,5)	99.05	96.30	98.62	97.65	99.51	**100**	**100**	99.75
	(1,3,5)	99.05	97.20	98.96	98.11	99.51	**100**	**100**	99.75
	(1,4,5)	99.05	**98.11**	99.31	**98.58**	99.51	**100**	**100**	99.75
	(2,3,5)	99.05	93.69	97.58	96.30	99.75	99.75	99.89	99.75
	(2,4,5)	99.05	96.30	98.62	97.65	99.75	**100**	**100**	**99.88**
	(3,4,5)	99.05	92.04	96.89	95.41	99.75	**100**	**100**	**99.88**
	(1,2,3,4)	99.05	97.20	98.96	98.11	99.51	99.75	99.89	99.63
	(1,2,3,4,5)	99.05	**98.11**	**99.31**	**98.58**	99.51	**100**	**100**	99.75

Table 6. Spread of brain tumor detection performance on testing 1

Performance (min – max)	**Model**	**Learning rate**			
		0.01	**0.001**	**0.0001**	**0.00001**
Acc (%)	tpCNN	93.65 - 96.45	94.16 - 97.21	93.65 - 97.72	94.42 - 97.21
	Voting $(t \geq n/2)$	95.69 - 97.21	94.42 - 97.46	95.43 - 96.95	95.94 - 97.72
	Voting $(t \geq 1)$	**97.46 - 98.98**	**97.46 - 99.24**	**97.21 - 98.98**	**97.21 - 98.98**
Sen (%)	tpCNN	98.10 - 99.05	99.05 - 99.05	93.65 - 97.72	98.10 - 99.05
	Voting $(t \geq n/2)$	**99.05 - 99.05**	99.05 - 99.05	95.43 - 96.95	**99.05 - 99.05**
	Voting $(t \geq 1)$	98.10 - 99.05	99.05 - 99.05	**97.21 - 98.98**	98.10 - 99.05
Pre (%)	tpCNN	81.25 - 88.89	82.54 - 91.23	81.25 - 92.86	83.20 - 91.96
	Voting $(t \geq n/2)$	86.67 - 91.23	83.20 - 92.04	85.95 - 90.43	87.39 - 92.86
	Voting $(t \geq 1)$	**92.04 - 98.10**	**92.04 - 98.11**	**91.23 - 97.20**	**91.23 - 98.10**
Spe (%)	tpCNN	91.70 - 95.50	92.39 - 96.54	91.70 - 97.23	92.73 - 96.89
	Voting $(t \geq n/2)$	94.46 - 96.54	92.73 - 96.89	94.12 - 96.19	94.81 - 97.23
	Voting $(t \geq 1)$	**96.89 - 99.31**	**96.89 - 99.31**	**96.54 - 98.96**	**96.54 - 99.31**
F-score (%)	tpCNN	89.27 - 93.69	90.04 - 94.98	89.27 - 95.85	90.43 - 94.93
	Voting $(t \geq n/2)$	92.44 - 94.98	90.43 - 95.41	92.04 - 94.55	92.86 - 95.85
	Voting $(t \geq 1)$	**95.41 - 98.10**	**95.41 - 98.58**	**94.98 - 98.11**	**94.98 - 98.10**

specificity, and F-score . However, there is no increase in the sensitivity value, as shown in Table 6.

The test results on the dataset of testing 2, tpCNN with input shape 1 and a learning rate 0.0001 yielded the best detection accuracy of 99.62%. In input shape 2 and input shape 3, tpCNN obtained the best detection accuracy at a learning rate of 0.00001, namely 99.85% and 99.39%, respectively. In input shape 4 and input shape 5, tpCNN obtained the highest accuracy of 99.08% and 99.62%, respectively, at a learning rate of 0.0001. From all these results,

Table 7. Spread of brain tumor detection performance on testing 2

Performance (min – max)	**Model**	**Learning rate**			
		0.01	**0.001**	**0.0001**	**0.00001**
Acc (%)	tpCNN	98.40 - 99.31	98.17 - 99.54	98.09 - 99.62	98.09 - 99.85
	Voting ($t \geq n/2$)	98.86 - 99.62	99.31 - 99.85	99.24 - 99.77	99.54 - 99.92
	Voting ($t \geq 1$)	**99.62 - 99.77**	**99.77 - 99.92**	**99.69 - 99.85**	**99.77 - 99.92**
Sen (%)	tpCNN	99.51 - 99.75	99.51 - 99.75	99.26 - 99.75	99.51 - 99.75
	Voting ($t \geq n/2$)	**99.75 - 99.75**	**99.75 - 99.75**	**99.75 - 99.75**	**99.75 - 99.75**
	Voting ($t \geq 1$)	99.51 - 99.75	99.51 - 99.75	99.26 - 99.75	99.51 - 99.75
Pre (%)	tpCNN	95.28 - 98.29	94.61 - 98.77	94.39 - 99.50	94.39 - 99.75
	Voting ($t \geq n/2$)	96.65 - 99.02	98.04 - 99.75	97.82 - 99.51	98.78 - 100
	Voting ($t \geq 1$)	**99.02 - 99.75**	**99.75 - 100**	**99.75 - 100**	**99.51 - 100**
Spe (%)	tpCNN	97.79 - 99.23	97.46 - 99.45	97.35 - 99.78	97.35 - 99.89
	Voting ($t \geq n/2$)	98.45 - 99.56	99.12 - 99.89	99.01 - 99.78	99.45 - 100
	Voting ($t \geq 1$)	**99.56 - 99.89**	**99.89 - 100**	**99.89 - 100**	**99.78 - 100**
F-score (%)	tpCNN	97.47 - 98.90	97.12 - 99.26	97.00 - 99.38	97.00 - 99.75
	Voting ($t \geq n/2$)	98.18 - 99.38	98.89 - 99.75	98.78 - 99.63	99.26 - 99.88
	Voting ($t \geq 1$)	**99.38 - 99.63**	**99.63 - 99.88**	**99.50 - 99.75**	**99.63 - 99.88**

tpCNN on the input shape 2 gives better detection results than the other input shapes.

In the voting scenario involving three tpCNN models, voting with $t \geq n/2$ (majority voting) resulted in the highest detection accuracy of 99.85% applied models (1,2,3) and (1,2,5) at a learning rate of 0.00001. The lowest accuracy in this scenario was 98.89% at a learning rate of 0.01. Whereas voting with $t \geq 1$ obtained the highest accuracy of 99.92% involved models (2,3,4), (2,4,5), and (3,4,5) at learning rates of 0.001 and 0.00001. At the same time, the lowest accuracy in this voting scenario was 99.62%, which involved model (2,3,4) at a learning rate of 0.01.

In voting involving four models (1,2,3,4), voting with the condition $t \geq n/2$ produced the best detection accuracy of 99.92% at a learning rate of 0.00001. While voting with the condition $t \geq 1$ obtained the highest tumor detection accuracy of 99.85% at a learning rate of 0.00001. For a voting scenario involving five models, voting with the condition $t \geq n/2$ achieved the best detection accuracy of 99.77% at learning rates of 0.001 and 0.0001. Meanwhile, voting $t \geq 1$ obtained the highest accuracy of 99.85% at learning rates of 0.001 and 0.00001.

Based on the results of testing 2, voting with $t \geq n/2$ conditions involving scenarios of three models, four models, or five tpCNN models shows an increase in accuracy in detection. However, the $t \geq 1$ voting proposed produces a generally better increase in detection accuracy with a minimum value of 99.62%, even though the maximum value for detection accuracy is the same, as shown in Table 4. Although the value of detection sensitivity at the learning rate 0.001 for voting $t \geq 1$ has not yet increased, in general, there is an increase in the value of precision, specificity, and F-score in each scenario. This scenario's precision and specificity values can reach 100%, as shown in Table 5. Judging from the performance dispersion value in detection at each learning rate, voting with the condition $t \geq 1$ in each scenario also shows a significant increase in accuracy, precision, specificity, and F-score . However, there is not yet an increase in sensitivity, as shown in Table 7.

5.2 Discussion

In this section, we investigate the performance of the proposed CNN model (tpCNN) with different input MRI image shapes as initial detection and the proposed voting method as final detection.

The tpCNN model, which involves three network paths, provides quite a variety of tumor detection results as an initial detection. In input shape 1 (64x64x3), the tpCNN model, tested at different learning rates, gave 95.18%-97.72% results. While on input shape 2, input shape 3, input shape 4, and input shape 5 obtained results of 93.65%-96.70%, 93.91%-95.45%, 94.42%-96.45%, and 93.65%-96.45%, respectively. The range of results shows intersections of results from each input shape but provides different minimum and maximum accuracy results. In input shape 1, the tpCNN model offers a better range of detection results than the others. Viewed based on the learning rate, tpCNN provides similar accuracy, as shown in Table 6 and Table 7. These results show that the influence of the input shape on tpCNN is more significant than the learning rate. When viewed from the number of parameters of the tpCNN model in the different input shapes,

Table 8. Comparison of the proposed method with the existing methods

No.	Authors	Methods	Acc (%)	
			Testing 1	Testing 2
1	K. Simonyan, et al. [43]	VGG16	94.67	99.47
2	K. Simonyan, et al. [43]	VGG19	96.19	99.54
3	K. He, et al. [44]	ResNet50	86.55	96.41
4	M. Sandler, et al.[45]	MobileNetV2	95.94	99.77
5	C. Szegedy, et al.[46]	InceptionV3	97.46	99.08
6	F. Chollet [47]	Xception	93.15	99.77
7	Proposed method	New Voting of tpCNN	**99.24**	**99.92**

tpCNN with input shape 4 (512x512x3) has more parameters than the others. However, the number of these parameters does not guarantee that the resulting tumor detection performance is comparable.

The test results with some different input shapes and learning rates in testing 1, the proposed tpCNN to get the highest accuracy in detecting brain tumors was 97.72%. These results identified tpCNN as having good performance because, in this study, we did not add a special pre-process that could improve tumor detection performance. To improve this performance, we added a voting process by considering several tpCNN decision results on some input shapes of MRI images. If the tpCNN with an input shape is a radiologist, the results of a diagnosis decision by considering several radiologists can strengthen its performance in detecting brain tumors. Voting with the condition $t \geq n/2$ is, in principle, a majority voting where the trial results in testing 1 give relatively small improvement results. tpCNN produced the best detection accuracy at a learning rate of 0.0001 of 93.65% - 97.72%, while voting with $t \geq n/2$ produced an accuracy of 95.94% - 97.72% at a learning rate of 0.00001. The maximum accuracy yielded by tpCNN is the same as voting $t \geq n/2$. However, voting $t \geq n/2$ can increase the minimum accuracy. Meanwhile, the proposed $t \geq 1$ voting method obtained an accuracy of 97.46% - 99.24% and is better than all. The $t \geq 1$ condition for voting is strict because if only one tpCNN with an input shape detects a brain tumor in the MRI image, the final detection indicates a tumor. This study's results are essential in early detection efforts before further examination. With this provision, voting with $t \geq 1$ can significantly increase the precision, the specificity, and the F-score and maintain high sensitivity.

The test results on the dataset of testing 2 (Table 7), tpCNN with different input shapes generally produce quite good detection performance based on the resulting performance dispersion. Voting with $t \geq n/2$ generally has improved performance better than tpCNN. However, voting with $t \geq 1$ gives the best result, even though voting $t \geq n/2$ is better in sensitivity. These results indicate that voting with $t \geq 1$can indirectly retrieve brain tumor features via tpCNN with specific input shapes, even though, at the same time, tpCNN with other input shapes cannot retrieve these features.

To determine the performance of the proposed method, we also compared it with the accuracy yielded by other well-known CNN models such as VGG16 [43], VGG19 [43], ResNet50 [44], MobileNetV2 [45], InceptionV3 [46], and Xception [47]. The model is pre-trained with the standard ImageNet dataset [48] to obtain initial weights. This process often solves classification problems involving small training image datasets. The model has two main structural parts, namely, the convolutional part and the classifier part. The convolutional section contains several convolution processes to extract the input image features, while the classifier section will classify these features into one of the target classes. Almost all of these models have a large number of parameters, thus they require more intensive computing.

The "MRI image-based brain tumor detection" case in this study has two classes, therefore, we apply the transfer learning technique by adding an output layer with two classes to all the models being compared. In the training process for all models, they are tested using the "Adam" optimizer with the "binary cross entropy" loss function. Meanwhile, the learning rate, the batch size, and the number of epochs used in the training process are 0.001, 16, and 100, respectively. Table 8 shows the results of tumor detection accuracy for all compared models.

Test results on dataset of testing 1, VGG16, VGG19, ResNet50, MobileNetV2, InceptionV3, and Xception yielded detection accuracy of 94.67%, 96.19%,86.55%, 95.94%, 97.46%, and 93.15%, respectively. These results show that InceptionV3 has better performance than other models. However, the proposed method, namely new voting of tpCNN, provided better results than InceptionV3 and others, with a tumor detection accuracy of 99.24%. Meanwhile, for testing on the dataset of testing 2, the proposed method also provided better results than the others, with a detection accuracy of 99.92%. The VGG 16, VGG19, InceptionV3, Xception, and MobileNetV2 are designed to handle small images, consequently, these models used filters on the first convolutional layer with a size of 3 × 3 to be able to find small patterns. At the same time, Resnet50 used a filter on the first convolutional layer with a size of 7 × 7 to find larger patterns, although the next convolution process used a filter size of 3 × 3. Brain tumor patterns vary widely, including relatively very small, medium, and large, thus using filter sizes representing these three sizes in convolution operations would be a better option. We used these three filter sizes in our proposed model. We approached the three filter sizes in the convolution process with sizes 3 × 3, 5 × 5, and 7 × 7, which work in parallel. However, the success of retrieving brain tumor features also depends greatly on the size of the input MRI image, therefore, our proposed model is implemented on a scheme involving several different input image shapes. Next, important features of the tumor can be extracted using the proposed voting method.

6. Conclusion

We have proposed CNN voting for brain tumor detection based on brain MRI images. The method is designed by forming several different input shapes. Tumor detection is carried out through two stages of detection. The first stage is the detection of tumors with varying input shapes using the proposed CNN model, and the second is the detection of the final using the proposed voting method. CNN in the proposed method has three network paths, each involving a convolution process with a different kernel size. The proposed voting method has strict provisions; if only one proposed CNN model detects a brain tumor in the MRI image, the final detection results will indicate a tumor. Testing with MRI image datasets, the proposed method yielded the best accuracy, sensitivity, precision, specificity, and F-score, respectively 99.24%, 99.05%, 98.11%, 99.31%, and 98.58% for small size and 99.92%, 99.75%, 100%, 100% and 99.88% for large size. With these results, the proposed method in this study has a high potential to assist radiologists in detecting tumors.

For clinical purposes, there are remaining opportunities to improve the performance of the proposed method in detecting brain tumors by involving all MRI sequences and planes.

Conflicts of interest

The authors declare no conflict of interest.

Author contributions

Conceptualization, Irwan Budi Santoso; methodology, Irwan Budi Santoso, software, Irwan Budi Santoso; validation, Irwan Budi Santoso, formal analysis, Irwan Budi Santoso, Shoffin Nahwa Utama, and Supriyono; investigation, Irwan Budi Santoso and Shoffin Nahwa Utama; resources, Irwan Budi Santoso; data curation, Irwan Budi Santoso and Shoffin Nahwa Utama; writing—original draft preparation, Irwan Budi Santoso; writing—review and editing, Irwan Budi Santoso, Shoffin Nahwa Utama, and Supriyono; supervision, Irwan Budi Santoso, Shoffin Nahwa Utama, and Supriyono; project administration, Irwan Budi Santoso; funding acquisition, Irwan Budi Santoso.

Acknowledgments

We want to thank the scientific group team of Informatics Engineering at Universitas Islam Negeri Maulana Malik Ibrahim Malang Indonesia for supporting the implementation of this research.

References

[1] S. Pereira, A. Pinto, V. Alves, and C. A. Silva, “Brain Tumor Segmentation Using Convolutional Neural Networks in MRI Images”, *IEEE Transactions on Medical Imaging*, Vol. 35, No. 5, pp. 1240–1251, 2016.

[2] World Health Organization, *Cancer*. Available online at: https://www.who.int/health-topics /cancer#tab=tab_1.

[3] S. Chatterjee, F. A. Nizamani, A. Nürnberger, and O. Speck, “Classification of brain tumours in MR images using deep spatiospatial models”, *Scientific Reports*, Vol. 12, No. 1, pp. 1–11, 2022.

[4] M. S. I. Khan, A. Rahman, T. Debnath, M. R. Karim, M. K. Nasir, S. S. Band, A. Mosavi, and I. Dehzangi, “Accurate brain tumor detection using deep convolutional neural network”, *Computational and Structural Biotechnology Journal*, Vol. 20, pp. 4733–4745, 2022.

[5] A. R. Loughan, F. J. Aslanzadeh, J. Brechbiel, G. Rodin, M. Husain, S. E. Braun, K. D. Willis, and A. Lanoye, “Death-related distress in adult primary brain tumor patients”, *Neuro-Oncology Practice*, Vol. 7, No. 5, pp. 498–506, 2020.

[6] P. Y. Wen, D. R. Macdonald, D. A. Reardon, T. F. Cloughesy, A. G. Sorensen, E. Galanis, J. DeGroot, W. Wick, M. R. Gilbert, A. B. Lassman, C. Tsien, T. Mikkelsen, E. T. Wong, M. C. Chamberlain, R. Stupp, K. R. Lamborn, M. A. Vogelbaum, M. J. V. D. Bent, and S. M. Chang, "Updated response assessment criteria for high-grade gliomas: Response assessment in neuro-oncology working group", *Journal of Clinical Oncology*, Vol. 28, No. 11, pp. 1963–1972, 2010.

[7] E. A. S. E. Dahshan, H. M. Mohsen, K. Revett, and A. B. M. Salem, "Computer-aided diagnosis of human brain tumor through MRI: A survey and a new algorithm", *Expert Systems with Applications*, Vol. 41, No. 11, pp. 5526–5545, 2014.

[8] J. Kang, Z. Ullah, and J. Gwak, "Mri-based brain tumor classification using ensemble of deep features and machine learning classifiers", *Sensors*, Vol. 21, No. 6, pp. 1–21, 2021.

[9] S. Asif, W. Yi, Q. U. Ain, J. Hou, T. Yi, and J. Si, "Improving Effectiveness of Different Deep Transfer Learning-Based Models for Detecting Brain Tumors From MR Images", *IEEE Access*, Vol. 10, pp. 34716–34730, 2022.

[10] S. Ahmad and P. K. Choudhury, "On the Performance of Deep Transfer Learning Networks for Brain Tumor Detection Using MR Images", *IEEE Access*, Vol. 10, pp. 59099–59114, 2022.

[11] S. Shanthi, S. Saradha, J. A. Smitha, N. Prasath, and H. Anandakumar, "An efficient automatic brain tumor classification using optimized hybrid deep neural network", *International Journal of Intelligent Networks*, Vol. 3, pp. 188–196, 2022.

[12] T. A. Jemimma and Y. J. Vetharaj, "A Survey on Brain Tumor Segmentation and Classification", *International Journal of Software Innovation*, Vol. 10, No. 1, 2022.

[13] M. Wang, "Multi-path Convolutional Neural Networks for Complex Image Classification", 2015. Available online at: http://arxiv.org/abs/1506.04701

[14] I. B. Santoso, Y. Adrianto, A. D. Sensusiati, D. P. Wulandari, and I. K. E. Purnama, "Ensemble Convolutional Neural Networks With Support Vector Machine for Epilepsy Classification Based on Multi-Sequence of Magnetic Resonance Images", *IEEE Access*, Vol. 10, pp. 32034–32048, 2022.

[15] I. B. Santoso and I. K. E. Purnama, "Epileptic EEG Signal Classification Using Convolutional Neural Networks Based on Optimum Window Length and FFT's Length", In: *Proc. of the 8th International Conference on Computer and Communications Management*, Association for Computing Machinery, New York, NY, USA, pp. 87–91, 2020.

[16] E. Arı and E. Taçgın, "Input Shape Effect on Classification Performance of Raw EEG Motor Imagery Signals with Convolutional Neural Networks for Use in Brain—Computer Interfaces", *Brain Sciences*, Vol. 13, No. 2, 2023.

[17] I. B. Santoso, Y. Adrianto, A. Sensusiati, D. Wulandari, and I. Purnama, "Epileptic EEG Signal Classification Using Convolutional Neural Network Based on Multi-Segment of EEG Signal", *International Journal of Intelligent Engineering and Systems*, Vol. 14, No. 3, pp. 160–176, 2021, doi: 10.22266/ijies2021.0630.15.

[18] A. Younis, L. Qiang, C. O. Nyatega, M. J. Adamu, and H. B. Kawuwa, "Brain Tumor Analysis Using Deep Learning and VGG-16 Ensembling Learning Approaches", *Applied Sciences*, Vol. 12, No. 14, 2022.

[19] N. Noreen, S. Palaniappan, A. Qayyum, I. Ahmad, M. Imran, and M. Shoaib, "A Deep Learning Model Based on Concatenation Approach for the Diagnosis of Brain Tumor", *IEEE Access*, Vol. 8, pp. 55135–55144, 2020.

[20] M. Rizwan, A. Shabbir, A. R. Javed, M. Shabbir, T. Baker, and D. A. J. Obe, "Brain Tumor and Glioma Grade Classification Using Gaussian Convolutional Neural Network", *IEEE Access*, Vol. 10, pp. 29731–29740, 2022.

[21] A. S. Musallam, A. S. Sherif, and M. K. Hussein, "A New Convolutional Neural Network Architecture for Automatic Detection of Brain Tumors in Magnetic Resonance Imaging Images", *IEEE Access*, Vol. 10, pp. 2775–2782, 2022.

[22] W. Jun and Z. Liyuan, "Brain Tumor Classification Based on Attention Guided Deep Learning Model", *International Journal of Computational Intelligence Systems*, Vol. 15, No. 1, pp. 1–9, 2022.

[23] S. Bhuvaji, A. Kadam, P. Bhumkar, S. Dedge, and S. Kanchan, "Brain Tumor Classification (MRI)", Available online at: https://www.kaggle.com/datasets/sartajbhuvaji/brain-tumor-classification-mri.

[24] N. Chakrabarty, "Brain MRI Images for Brain Tumor Detection", Available online at: https://www.kaggle.com/datasets/navoneel/brain-mri-images-for-brain-tumor-detection.

[25] A. Hamada, "Br35H Brain Tumor Detection 2020", Available online at:

https://www.kaggle.com/datasets/ahmedhamada0/brain-tumor-detection.

[26] M. Nickparvar, “Brain Tumor MRI Dataset”, Available online at: https://www.kaggle.com/datasets/masoudnickparvar/brain-tumor-mri-dataset.

[27] J. L. Ba, J. R. Kiros, and G. E. Hinton, “Layer Normalization”, 2016, Available online at: http://arxiv.org/abs/1607.06450.

[28] Z. S. Chadaj, T. D. Bel, L. Blanchet, A. Baidoshvili, D. Vossen, J. V. D. Laak, and G. Litjens, “Impact of rescanning and normalization on convolutional neural network performance in multi-center, whole-slide classification of prostate cancer”, *Scientific Reports*, Vol. 10, No. 1, pp. 1–14, 2020.

[29] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks”, *Communications of the ACM*, Vol. 60, No. 6, pp. 84–90, 2017.

[30] K. He, X. Zhang, S. Ren, and J. Sun, “Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification”, In: *Proc. of 2015 IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile, pp. 1026–1034, 2015.

[31] Y. Yuan, G. Xun, K. Jia, and A. Zhang, “A Multi-View Deep Learning Framework for EEG Seizure Detection”, *IEEE Journal of Biomedical and Health Informatics*, Vol. 23, No. 1, pp. 83–94, 2019.

[32] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks”, In: *Proc. of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS)*, Sardinia, Italy, Vol. 9, pp. 249–256, 2010.

[33] V. Dumoulin and F. Visin, “A guide to convolution arithmetic for deep learning”, pp. 1–31, 2016, Available online at: http://arxiv.org/abs/1603.07285

[34] J. Nagi, F. Ducatelle, G. A. D. Caro, D. Ciresan, U. Meier, A. Giusti, F. Nagi, J. Schmidhuber, and L. M. Gambardella, “Max-pooling convolutional neural networks for vision-based hand gesture recognition”, In: *Proc. of 2011 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, Kuala Lumpur, Malaysia, pp. 342–347, 2011.

[35] R. Nirthika, S. Manivannan, A. Ramanan, and R. Wang, “Pooling in convolutional neural networks for medical image analysis: a survey and an empirical study”, *Neural Computing and Applications*, Vol. 34, No. 7, pp. 5321–5347, 2022.

[36] J. Jin, A. Dundar, and E. Culurciello, “Flattened convolutional neural networks for feedforward acceleration”, In: *Proc. of the International Conference on Learning Representations(ICLR)*, San Diego, CA, USA, pp. 1–11, 2015.

[37] A. B. Malayeri and M. B. Khodabakhshi, “Concatenated convolutional neural network model for cuffless blood pressure estimation using fuzzy recurrence properties of photoplethysmogram signals”, *Scientific Reports*, Vol. 12, No. 1, pp. 1–14, 2022.

[38] A. Patil and M. Rane, “Convolutional Neural Networks: An Overview and Its Applications in Pattern Recognition”, *Smart Innovation, Systems and Technologies*, Vol. 195, pp. 21–30, 2021.

[39] C. M. Bishop, “Pattern Recognition and Machine Learning”, *Springer*, New York, NY, 2006.

[40] C. Zhang and Y. Ma, “Ensemble machine learning: Methods and applications”, *Springer New York Dordrecht Heidelberg Londom*, 2012.

[41] J. D. Rodriguez, A. Perez, and J. A. Lozano, “Sensitivity Analysis of k-Fold Cross Validation in Prediction Error Estimation”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 32, No. 3, pp. 569–575, 2010.

[42] S. R. Dubey, S. Chakraborty, S. K. Roy, S. Mukherjee, S. K. Singh, and B. B. Chaudhuri, “diffGrad: An Optimization Method for Convolutional Neural Networks”, *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 31, No. 11, pp. 4500–4511, 2020.

[43] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition”, pp. 1–14, 2014, Available online at: http://arxiv.org/abs/1409.1556

[44] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition”, In: *Proc. of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, pp. 770–778, 2016.

[45] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks”, In: *Proc. of 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, pp. 4510–4520, 2018.

[46] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the Inception Architecture for Computer Vision”, In: *Proc. of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, pp. 2818–2826, 2016.

[47] F. Chollet, “Xception: Deep learning with depthwise separable convolutions”, In: *Proc. of 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, pp. 1800–1807, 2017.

[48] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. F. Fei, “ImageNet: A large-scale hierarchical image database”, In: *Proc. of 2009 IEEE Conference on Computer Vision and Pattern Recognition*, FL, USA, pp. 248-255 2009.