

## Article

# An Ensemble Classification Method for Brain Tumor Images Using Small Training Data

Dat Tien Nguyen, Se Hyun Nam , Ganbayar Batchuluun, Muhammad Owais  and Kang Ryoung Park  \*

Division of Electronics and Electrical Engineering, Dongguk University, 30 Pildong-ro, 1-gil, Jung-gu, Seoul 04620, Republic of Korea

\* Correspondence: parkgr@dongguk.edu

**Abstract:** Computer-aided diagnosis (CAD) systems have been used to assist doctors (radiologists) in diagnosing many types of diseases, such as thyroid, brain, breast, and lung cancers. Previous studies have successfully built CAD systems using large, annotated datasets to train their models. The use of a large volume of training data helps these CAD systems to collect rich information for application in the diagnosis process. However, a large amount of training data is sometimes unavailable for training the models, such as for a new or less common disease and diseases that require expensive image acquisition devices. In such cases, conventional CAD systems are unable to learn their models efficiently. As a result, diagnostic performance is reduced. In this study, we focus on dealing with this problem; thus, our classification method can enhance the performance of conventional CAD systems based on the ensemble model of a support vector machine (SVM), multilayer perceptron (MLP), and few-shot (FS) learning network when working with small training datasets of brain tumor images. Through experiments, we confirmed that our proposed method outperforms conventional deep learning-based CAD systems when working with a small training dataset. In detail, we verified that the lack of training data led to the reduction of classification performance. In addition, we enhanced the classification accuracy from 3% to 10% compared to previous studies that used the SVM-based classification method or fine-tuning of a convolutional neural network (CNN) using two public datasets.



**Citation:** Nguyen, D.T.; Nam, S.H.; Batchuluun, G.; Owais, M.; Park, K.R. An Ensemble Classification Method for Brain Tumor Images Using Small Training Data. *Mathematics* **2022**, *10*, 4566. <https://doi.org/10.3390/math10234566>

Academic Editor: Jakub Nalepa

Received: 13 October 2022

Accepted: 30 November 2022

Published: 2 December 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

With the development of imaging technology, many organs of the human body can be made easily visible on images, which helps to diagnose many kinds of diseases without the need to perform operations [1–3]. Due to this advantage, imaging techniques are widely used as high-quality and cheap diagnostic methods compared to other diagnostic methods. Many imaging techniques have been developed to suit different organ types and diseases. For example, X-ray imaging was one of the earliest imaging techniques used to capture images of the lung and bone [2]. Using X-ray images, doctors can assess the lung cancer condition or determine the position of the broken bone [2]. Ultrasound imaging techniques are currently used to capture images of the breast or thyroid region to detect and evaluate the condition of breast and thyroid cancers [3]. Computed tomography (CT) or magnetic resonance imaging (MRI) are used to capture brain images [1,4]. Using these images, doctors (radiologists) can diagnose the type and condition of the disease and determine a more accurate treatment method based on their experiences. However, the disadvantage of this diagnosis and treatment method is that doctors (radiologists) must perform every step in the diagnosis process based on their knowledge and experience of a specific type of

disease. Therefore, diagnostic performance varies and depends heavily on the personal ability of a doctor (radiologist).

Computer-aided diagnosis (CAD) systems have been widely developed and used to assist doctors and to enhance the performance of diagnosis and treatment processes. This technique uses a computer program to process the captured images of organs and provides suggestions to doctors during the diagnosis and treatment process. The advantage of this method is that it can be trained using large amounts of data with the help of experts (experienced doctors). Therefore, performance is not affected by the personal ability of a doctor. In addition to these advantages, the design and development of CAD systems face several difficulties, such as the performance of the processing algorithm and data collection.

Brain diseases, such as brain tumors [5], brain strokes [6], and brain disorders [7], are important problems in human health. Brain tumors are the main cause of cancer-related death [5], whereas brain stroke also causes death and disability in elderly people with health problems, such as hypertension, abundant utilization of liquor, diabetes, cholesterol, and smoking [6]. Therefore, the classification of brain tumor/stroke images are important for the diagnosis and treatment of brain diseases. Studies on CAD systems for brain tumor classification problems are mainly based on either handcrafted image feature extraction, followed by a classification method or a deep learning network using a convolutional neural network (CNN).

### 1.1. Handcrafted Feature-Based Methods

In the first category, previous studies mainly used expert design feature extraction methods to extract information from brain images and classified the input images into predefined categories using either classification or distance measurement (image retrieval) methods. In a study by Jun Cheng et al. [8], they used the Fisher vector representation (FV) method to extract brain image features for several subregions. With the extracted FV, they can retrieve brain tumor images in a dataset using the similarity distance measurement between two FVs. Similarly, Huang et al. [9] used a bag-of-words model to successfully retrieve brain tumor images. Ismael et al. [10] used a 2D discrete wavelet transform and 2D Gabor filtering technique to extract statistical information from brain images, such as the mean, variance, skewness, and contrast, to form a feature vector that represents the content of the brain images. Finally, they used a multilayer perception network (MLP) to classify the input brain images into several predefined categories of brain images. Gurbina et al. [11] also used different types of wavelets transform to extract information from images and a support vector machine (SVM) to identify whether a brain image contains a tumor. Zaw et al. [12] used the naïve Bayes classification method to classify brain images based on eight region properties (area, perimeter, eccentricity, equivalent diameter, solidity, convex area, major axis length, and minor axis length) and three intensity features (maximum, mean, and minimum) of the extracted brain tumor region. Minz et al. [13] extracted gray level co-occurrence matrix (GLCM) features from brain images and used the Adaboost classification algorithm to classify an input brain image into cancerous or non-cancerous categories. While these methods can classify brain images into predefined categories, they use expert-design feature extractors (statistical, Fisher vector, wavelet transform, Gabor filter, and GLCM). As a result, the extracted image features capture only limited aspects of brain images, which can result in the low performance of classification systems.

### 1.2. Deep Feature-Based Method

Deep learning-based methods have recently been used to deal with the detection and classification of brain tumor images. Many deep learning-based systems have been proposed for brain tumor image segmentation or the classification problem. For example, Chen et al. [14] proposed a brain tumor segmentation network using a combination of the convolutional neural network (CNN) and transformer architecture, namely CSU-Net. This research yielded higher segmentation accuracy than other segmentation methods, such as the 3DU-Net network.

To overcome the limitations of handcrafted-based brain tumor classification methods, deep learning-based classification methods have recently been used. In a study by Abiwinanda et al. [5], they proposed the use of a convolutional neural network (CNN) to classify three types of brain tumors, including glioma, meningioma, and pituitary. They validated the classification performance using CNNs at various depths. As a result, they showed that a CNN that consists of two convolution layers and one dense (fully connected layer) layer outperforms other network architectures in their experiments. To reduce the number of network parameters of a CNN, Isunuri et al. [15] proposed the use of separable convolution in CNNs for brain tumor classification problems. In several other studies, Swati et al. [16], Kumar et al. [17], and Deepak et al. [18] used a fine-tuning method to train a CNN for brain tumor image classification problems. By using the pretrained CNN models (such as AlexNet, VGG16, VGG19, residual network, and GoogleNet network architectures that were successfully trained on the ImageNet dataset for a general image classification problem), they successfully trained the classification model using a small amount of training data that is difficult to perform compared to training the network from scratch. Furthermore, Alanazi et al. [19] proposed a CNN-based classification method based on a fine-tuning procedure using two brain tumor datasets. In this approach, the authors first trained a CNN on the first dataset. The trained CNN model was then fine-tuned using the second dataset to associate the knowledge of the two datasets. They showed that the fine-tuning approach is efficient for the development of brain tumor image classification systems. Bodapati et al. [20] used two pretrained CNNs (Inception ResNet and Xception) to extract deep image features. Using two different CNN architectures, the authors showed that they can extract two different sets of feature representations. Finally, they proposed a fusion method based on a multi-layer perceptron (MLP) network to combine the extracted image features for the classification problem. To extract image features at different image scales, Togacar et al. [21] proposed the use of image features at various stages of a CNN. The shallow convolution layers help to extract low-abstract image features, whereas the deep convolution layers help to extract highly abstract image features. By combining low- and high-abstract image features, they showed that their proposed method is efficient for brain tumor image classification. In a similar approach, Diaz-Pernas et al. [22] proposed a multiscale CNN that processes input images using several subnetworks with different-sized convolution kernels to extract multiscale image features. They confirmed that multiscale features are efficient for brain tumor classification. Kang et al. [23] recently proposed an ensemble approach for the brain tumor classification problem. For this purpose, they first used three different pretrained CNN models, namely, DenseNet, InceptionNet, and residual-based networks, to extract image features. With the extracted image features, they applied several classification methods, such as support vector machines (SVMs), to the combined concatenated features to classify input images into several predefined categories of brain tumor images. By using pretrained CNNs that were successfully trained on a large dataset (ImageNet dataset), Kang's method does not need to train the CNN again while utilizing the power of deep features. A similar study conducted by Deepak et al. [24] confirmed the efficiency of deep features and SVM-based classifiers in brain tumor classification systems. Kesav et al. [25] proposed a combination of detection and classification networks for brain tumor detection and the classification problem. They first constructed a two-channel CNN network to efficiently classify brain images into either tumor or non-tumor images. After that, a region-based CNN (RCNN) network is used to identify the tumor region. Most recently, Chatterjee et al. [26] proposed spatio-spatial models for classifying the 3D scan of brain images into different types of brain tumors by learning information in both spatial and temporal spaces.

While these deep learning-based studies have shown that they successfully classify brain tumor images into predefined categories, they require a large amount of training data to train the classification network, which is normally difficult to obtain for medical image processing. Therefore, their performance will be significantly reduced in the case of less data being available for training. In addition, the classification performance can be affected

by the architecture of the CNN. Table 1 summarizes previous studies on the brain tumor image classification problem in comparison to our proposed method.

**Table 1.** Summary of previous studies on brain tumor image classification in comparison to our proposed method.

Category	Method	Strength	Weakness
Handcrafted feature-based method	Extract image features by handcrafted methods [8–13]	Image features are extracted by expert knowledge that are suitable for the problem	Performance is limited as the image features are not optimal.
Deep feature-based method	<ul style="list-style-type: none"> <li>- CNN-based network [5]</li> <li>- CNN-based with separable convolution [15]</li> <li>- Fine-tuned deep CNN networks [16–22]</li> <li>- SVM-based classification based on deep feature extraction from pretrained CNNs [23,24]</li> <li>- Deep spatiotemporal models for 3D volumetric MRI image classification [26]</li> </ul>	<p>High performance can be obtained when training with large amount of training data</p>	<ul style="list-style-type: none"> <li>- Requires strong hardware equipment to perform training procedure.</li> <li>- Takes time to train network with large amount of data</li> </ul>
	Proposed Method: Combines classification results of SVM-, MLP-, and FS-based networks	<ul style="list-style-type: none"> <li>- Uses multiple CNN architectures for feature extraction</li> <li>- Ensemble of SVM, MLP, and FS networks</li> <li>- Requires fewer images for training network</li> </ul>	<ul style="list-style-type: none"> <li>- Complex architecture</li> <li>- Takes longer time to process than previous methods</li> </ul>

The rest of our paper is organized as follows. In Section 2, we highlight the main contribution of our work. In Section 3, we present our proposed method in detail. Using the proposed method, we performed experiments with two public datasets to evaluate the performance of brain tumor image classification using a small training dataset and the experimental results are presented in Section 4. In Section 5, we discuss our experimental results presented in Section 4. Finally, the conclusion of our work is presented in Section 6.

## 2. Contributions of Our Study

In this study, we focused on designing a brain tumor image classification system when small amounts of data are available for training. To overcome the limitations of previous studies, as well as to deal with small training dataset situations, we propose an ensemble deep learning-based classification network for brain tumor image classification problems. Our proposed method is novel in the following four ways compared with previous studies:

- Our proposed method is designed to address the problem of when a large number of brain tumor images is not available for training models.
- We designed a deep learning-based method to measure the distance between two images such that we can find the most similar image to the input image. We called this network a few-shot (FS) learning network because it can be trained using a small annotated dataset and uses few-shot learning technology to measure the similarity between the images.
- We combined the classification outputs of SVM, MLP, and FS methods as ensemble models using three fusion rules, including SVM, weighted-SUM, and weighted-PRODUCT, to enhance the performance of the individual method.
- We constructed our algorithm with a publicly available, pretrained model for fair comparison with our work [27].

## 3. Proposed Method

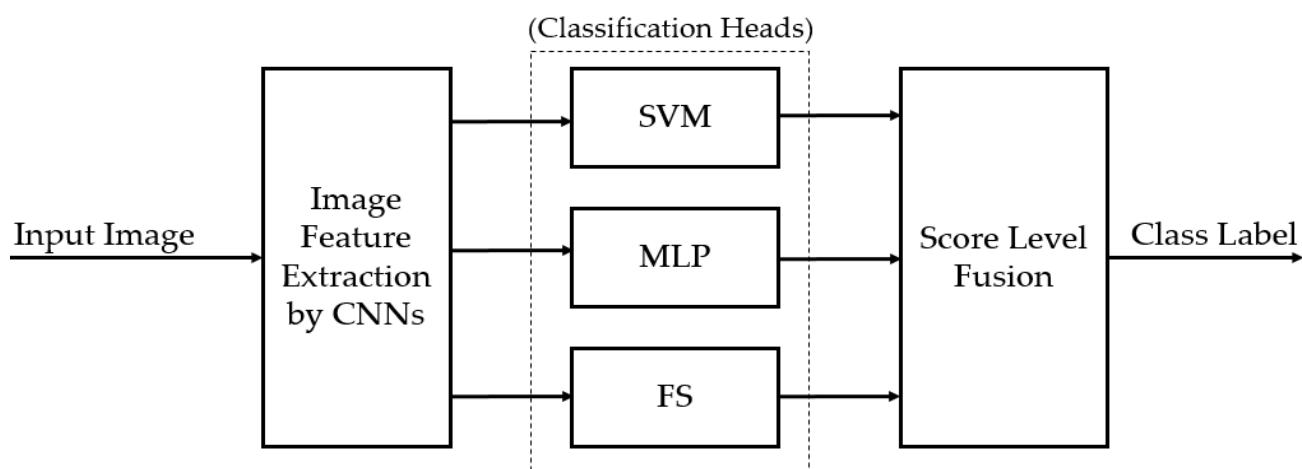
### 3.1. Datasets

In this study, we proposed a brain image classification system using the MRI imaging model. For this purpose, we used two public MRI datasets: BT-Small and BT-Large, to

evaluate the performance of our proposed method. The BT-Small dataset was downloaded from the Kaggle website [28], which contains 253 brain images, out of which 155 images were marked with tumors, and the remaining 98 images were marked without tumors (non-tumors). The second dataset is a public dataset that can be downloaded from its official website [29] and has been used in previous studies [8,30]. This dataset contains a total of 3064 T1-weighted images with three different types of brain tumors: Meningioma, glioma, and pituitary. We named these datasets as the BT-Small and BT-Large for convenience. Images in the BT-Small dataset are stored in jpeg file format with varied image resolution, whereas the images in the BT-Large dataset are stored in the Matlab file format with the fixed size of 512-by-512 pixels. As we are dealing with the training dataset that is relatively small, we performed several data augmentation methods, such as random cropping and resizing and flipping, on the input images in the training dataset. In detail, the flipping method yields an additional three flipped images (left-right, and up-down direction) for each input image; whereas we randomly cropped 90% of original images to make an augmented image using the cropping and resizing method. In total, we made about 23 augmented images as we repeated the cropping and resizing procedure 20 times for each image in the training dataset.

### 3.2. Overview of the Proposed Method

In Figure 1, we depict the flow chart of our proposed method for a small training-data-based brain tumor classification system. As shown in this figure, the proposed method consists of three classification heads: Support vector machines (SVMs), a MLP, and a FS network. As we are working with small training datasets (the number of training images is from 5 to 10), it is difficult to train deep CNNs or FS networks from scratch or by fine-tuning. Therefore, for the inputs to all classification heads, we used pretrained CNN models that were successfully trained on the ImageNet dataset to extract deep features. As the pretrained CNN models were trained on a large RGB color image dataset (ImageNet dataset), they are efficient to extract texture information from images such as blobs, line, and other complex texture features. While the MRI brain images are different from RGB color images in the ImageNet dataset, they also contain texture features such as blobs, lines, etc. Therefore, we designed our classification method in which the pretrained CNN networks are used only for image feature extraction. While it is not the best way to extract efficient image features from MRI images, it is acceptable as it was successfully used in previous study by Kang et al. [23] for the brain tumor image classification problem. This method helps our proposed method extract sufficient features from the input image without a training procedure. Descriptions of the SVM, MLP, and FS methods are given in Sections 3.4–3.6.

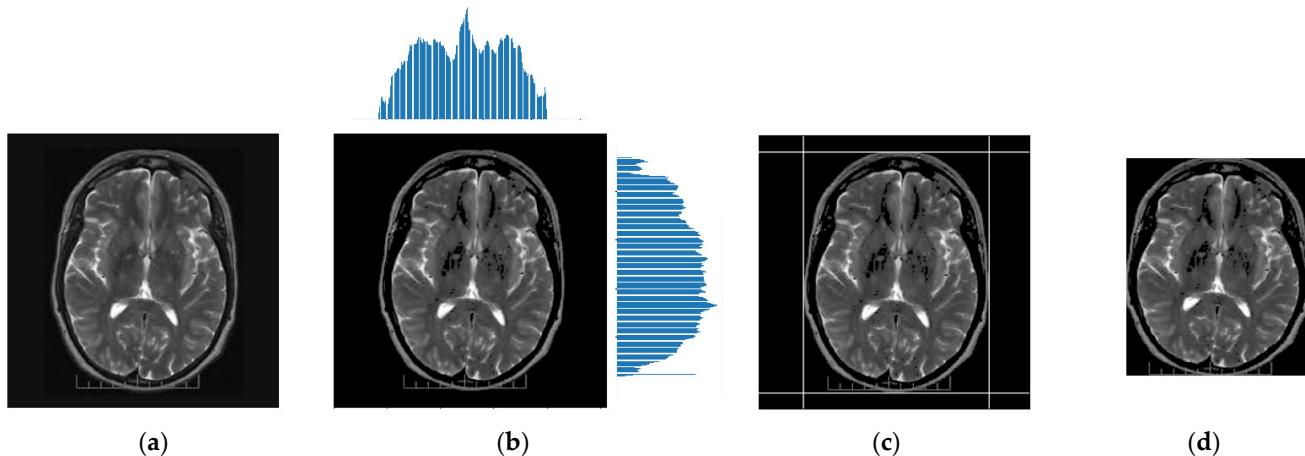


**Figure 1.** Overview of our proposed method.

As each classification method has advantages and disadvantages in the classification task, we further combined the classification results of the SVM, MLP, and FS methods using score-level fusion to enhance the classification accuracy of each method. In our proposed method, we proposed and evaluated three combination methods: SVM, weighted-SUM, and weighted-PRODUCT rules. Detailed descriptions of these fusion methods are provided in Section 3.7.

### 3.3. Preprocessing of Input Images

A captured brain image typically contains two parts: A brain and a background region, as shown in Figure 2a. As the background region contains no information for the brain tumor image classification problem, it should be removed before other processing steps. This step is important because it not only helps to remove non-information regions, but also helps to localize the brain region. With the localized brain region, we then performed a size normalization step to create a uniform brain region for our proposed method.



**Figure 2.** Example of preprocessing steps to extract the brain region image in our proposed method: (a) Captured brain image, (b) image after background removal with gray profiles in horizontal and vertical directions, (c) image with estimated brain ROI, and (d) final brain ROI image.

According to our observations, the background region of the brain tumor images, normally composed of darker pixels than that of the brain region (foreground region) that contains pixels in brighter. Based on this observation, we developed a simple but efficient method for brain region localization, as shown in the algorithm presented in Algorithm 1. An example of the localization results of a brain tumor image is shown in Figure 2 for demonstration purposes. As shown in this figure, we can efficiently localize the brain region of an input-captured brain image. In addition, we created a uniform brain image by scaling the detected brain regions to a fixed size of  $256 \times 256$  pixels. This final image is then inputted to our proposed method for feature extraction and classification, as shown in Figure 1.

In Algorithm 1, we used the size-threshold value to remove small-sized and dark objects from brain images. The optimal threshold was experimentally selected from training data.

We used two public MRI datasets in our experiments, i.e., BT-Small and BT-Large, as presented in Section 3.1. In these datasets, the brain images were captured in the axial plane. While there could exist a small misalignment among images, the orientation of the brain is not strongly varied. Therefore, our brain tumor region detection algorithm does not detect the brain orientation. As shown in Figure 2a, the background regions of brain MRI images mostly contain darker pixels compared to those in the brain regions. Based on this characteristic, we first classified the brain MRI image into two classes of the background and brain regions using a threshold value, as shown in Algorithm 1. This threshold can be calculated using some adaptive thresholding methods, such as the Otsu's method or experimentally decided based on characteristics of images. In our study, we used the

Otsu's method to adaptively detect the background region. As shown in Algorithm 1, we accumulated the gray profile in the horizontal and vertical directions to find the most concentrated region where the brain appears. Therefore, although there could exist some individual white pixels due to an error of the segmentation method, our algorithm can easily be removed during brain region estimation.

---

**Algorithm 1:** Preprocessing of brain tumor image
 

---

Input: A captured brain tumor image ( $I$ ) with size  $H \times W$   
 Steps:

- 1 Image binarization to completely force the background pixels to zeros using Equation (1).

$$M_I = I > threshold \quad (1)$$

- 2 Image masking by multiplying the original image with the binarized image using Equation (2).

$$I = I \times M_I \quad (2)$$

- 3 Accumulate the gray density levels (gray profile) along horizontal and vertical directions, i.e., PH and PV using Equations (3) and (4).

$$PH_j = \left\{ \frac{1}{W} \sum_{i=1}^W I_{ji} \right\} \text{ for } j = 0, \dots, H \quad (3)$$

$$PV_j = \left\{ \frac{1}{H} \sum_{i=1}^H I_{ij} \right\} \text{ for } j = 0, \dots, W \quad (4)$$

- 4 Find the left and right boundaries of each direction on gray profiles (PH and PV) by finding the first and last positions which have gray density levels greater than a size-threshold value.

- 5 Normalize the detected brain region image to a fixed size of  $256 \times 256$  pixels.

Output: Normalized  $256 \times 256$ -pixel image

---

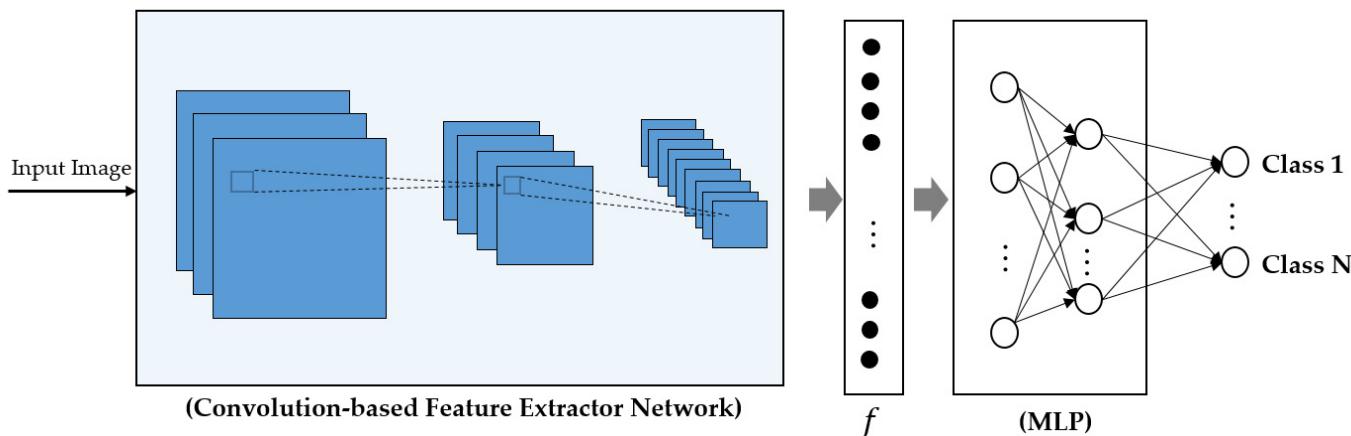
Of the two public datasets used in our experiments, the BT-small dataset contains various sizes of MRI images, whereas the BT-large dataset contains a fixed size of 512-by-512 pixels. As shown in Figure 2a, the brain region is normally much smaller than the original size of the brain image. In addition, our proposed method uses pretrained CNN models (Residual, Inception, and Dense network) for image feature extraction. These models are initially designed to accept the input image with a size of roughly 256-by-256 pixels, and the use of a larger size image can cause inefficiency in the processing speed. Due to these reasons, we resized the brain region to the fixed size of 256-by-256 pixels in our experiments.

### 3.4. Classification Based on MLP Network

Inspired by the success of the CNN for classification problems, our study also uses a CNN for brain image classification. The methodology of a CNN-based classification system is depicted in Figure 3 [31–37]. This technique uses a convolution operation with a trainable convolution kernel to extract efficient image features and a conventional multi-layer perceptron (MLP) network for the classification problem.

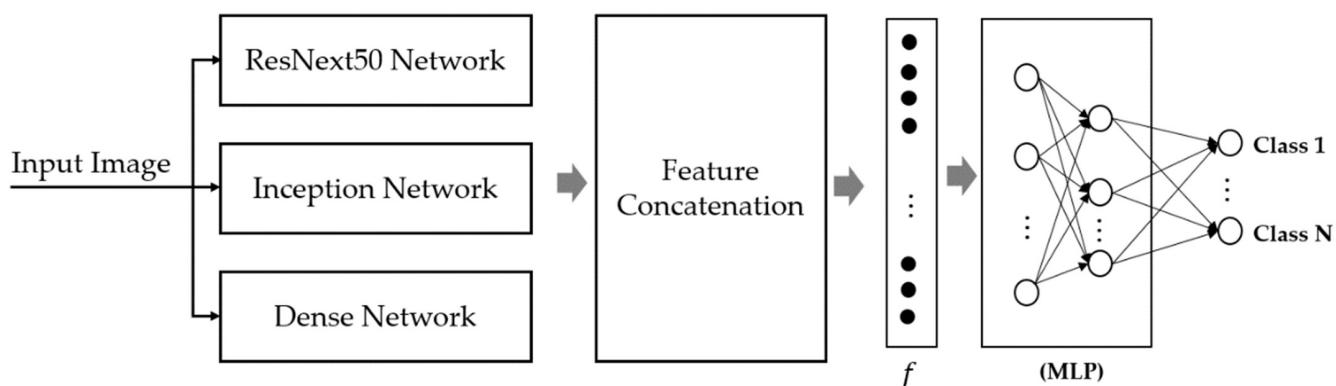
However, as we are dealing with a small training dataset, it is insufficient for training a CNN-based classification network from scratch because the CNN requires a large amount of training data to learn its parameters. Instead, we use CNN networks such as image feature extractors to extract efficient image features. This method has been successfully used in previous studies on image-based classification systems [23]. Based on the results obtained by Kang et al. [23], we used a multiple network architecture for image feature ex-

traction in our study, including: The residual network [35] (ResNeXt50 network), inception network [33] (Inception V3 network), and dense [34] (DenseNet169 network) network.



**Figure 3.** Methodology of the CNN-based method for the classification problem.

Based on the three popular network architectures, we designed our classification network using the extracted image features, as shown in Figure 4. As shown in this figure, we use a multi-layer perceptron network (MLP) to learn a classifier for brain tumor image classification problems using the image features extracted by three pretrained CNN models, including ResNext50, Inception, and Dense networks. While it takes longer for processing input images using multiple networks than using a single network, the use of three pretrained networks for image feature extraction helps to extract richer information than the use of a single network that can help to increase the classification performance of the overall system. An MLP network with a small number of neurons and layers was designed. Therefore, we do not need to learn many network parameters as we must when training a conventional CNN from scratch. In addition, we can utilize the power of the CNN by using pretrained networks as image feature extractors.



**Figure 4.** Our design for an MLP-based classification network for a small training dataset brain tumor classification problem.

The detailed description of the MLP-based network used in our study is given in Table 2. As shown in this table, we used three pretrained CNN models, including ResNeXt50, InceptionV3, and DenseNet169 network for image feature extraction. The image features are obtained at the output of the last convolution layer of each CNN model after a global average pooling operator. As a result, we obtained three feature vectors in a 4096-, 4096-, and 1664-dimensional space for the cases of using ResNeXt50, InceptionV3, and DenseNet169 network, respectively. These feature vectors are then concatenated to form a 5760-dimensional feature vector. Finally, a three-layer MLP network with 512, 1024,

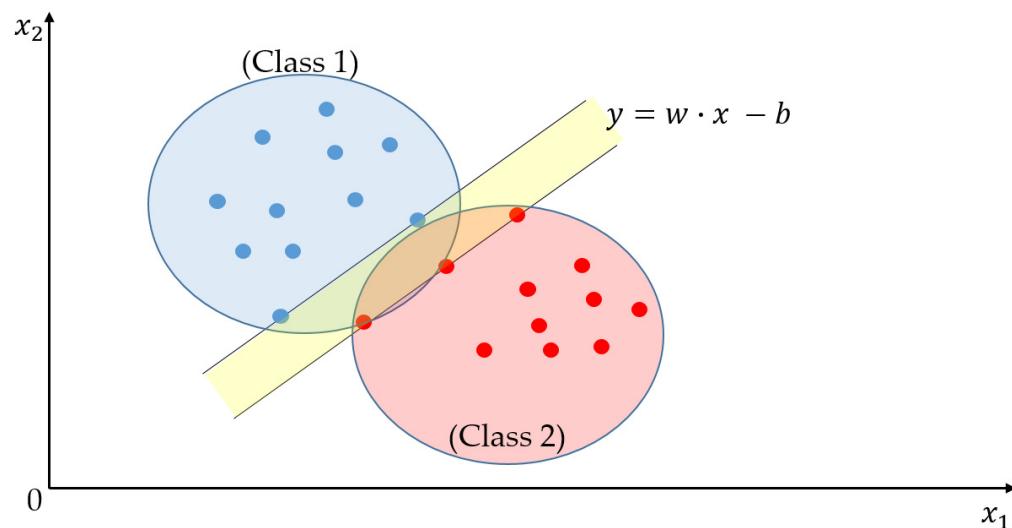
and C neurons was used to classify the input images into predefined categories using the extracted image features, as shown in Table 2. In this table, C indicates the number of predefined categories of brain images. In our experiments with the BT-Small dataset, C takes the value of two; whereas C takes the value of three in our experiments with the BT-Large dataset. As we used pretrained CNN models for image feature extraction, we froze the convolution layers of these CNN models in our experiments. As a result, the number of trainable parameters of these models are zero, as shown in Table 2.

**Table 2.** Detailed description of the MLP-based classification network in our study (C means the number of classes).

Layer	Output Dimension	#Trainable Parameters	
ResNeXt50	$1 \times 4096$	0	
InceptionV3	$1 \times 4096$	0	
DenseNet269	$1 \times 1664$	0	
Concatenation	$1 \times 5760$	0	
Dense Layer 1	$1 \times 512$	2,949,632	
MLP Layer	Dense Layer 2	$1 \times 1024$	525,312
	Dense Layer 3	$1 \times C$	$C \times 1024 + 2$

### 3.5. Classification Based on SVMs

The support vector machine (SVM) is a well-known and efficient method for classification and regression problems that are based on the selection of support vectors [38]. Suppose we have a training dataset of n samples of two classes, and each training sample is represented as a feature vector in an m-dimensional space. Then, the SVM method learns to select a subset of k feature vectors (called support vectors) to construct a hyperplane with the largest margin by which n samples are efficiently separated, as shown in Figure 5.



**Figure 5.** SVM method for finding the hyper-plane for the classification problem.

For a two-class classification problem, the classifier is selected by learning the parameters that satisfy Equation (5).

$$f(x) = \text{sign}(\sum_{i=1}^k a_i y_i K(x, x_i) + b) \quad (5)$$

where  $x_i$  and  $y_i$  denote the selected support vectors and their corresponding class labels, respectively;  $a_i$  and  $b$  denote the parameters of the classifier; and  $K()$  is a kernel function that is used to transform the input feature vector into a higher dimensional space in which the data samples can be linearly separable. The kernel function is normally useful when the original training data samples are not completely separable in the original data space but are more separated in a higher dimensional space. There are four popular kernel functions that have been widely used in applications: Linear, radial basis function (RBF), polynomial (Poly), and sigmoid functions, as shown in Equations (6)–(9). In our experiments, we evaluated classification performance using all four kernels. The detailed experimental results are presented in Section 4. In the case of a multiclass classification problem, the problem is simplified to a two-class classification problem using either a one-versus-one or one-versus-rest refinement strategy. In Equations (6)–(9),  $\gamma$ ,  $r$ , and  $deg$  are the transformation parameters and can be selected using a grid-search algorithm to find the best parameters for a given dataset.

$$\text{Linear Kernel} \quad K(x_i, x_j) = x_i^T x_j \quad (6)$$

$$\text{RBF Kernel} \quad K(x_i, x_j) = e^{-\gamma \|x_i - x_j\|^2} \quad (7)$$

$$\text{Polynomial Kernel} \quad K(x_i, x_j) = (\gamma x_i^T x_j + r)^{deg} \quad (8)$$

$$\text{Sigmoid Kernel} \quad K(x_i, x_j) = \tanh(\gamma x_i^T x_j + r) \quad (9)$$

We used three pretrained CNNs, including ResNext50 [35], Inception [33], and DenseNet [34], to extract the image features of the input images. With these image features, we performed SVM classification on the combined features using four SVM kernels mentioned in Equations (6)–(9). As shown in these equations, there are trainable parameters that need to be learnt to obtain SVM classifiers. To obtain the optimal parameters, we used a grid search method provided in the scikit-learn library [39] to find the best possible values.

### 3.6. Classification Based on FS Network

#### 3.6.1. Few-Shot Learning Technique

The few-shot learning technique is an efficient method that aims to recognize samples from very few labeled data. By definition, the few-shot learning technique is a relation network (RN) that consists of two components: An embedding module that is responsible for image feature representation and a relation module that is responsible for finding the relation (similarity) between two samples [40]. Suppose that we have a dataset containing images in  $N$  classes. Then, the embedding module ( $f_e$ ) is used to extract the feature map  $f_e(x_i)$  of the images in the dataset, as shown in Equation (10).

$$E_i = f_e(x_i) \quad (10)$$

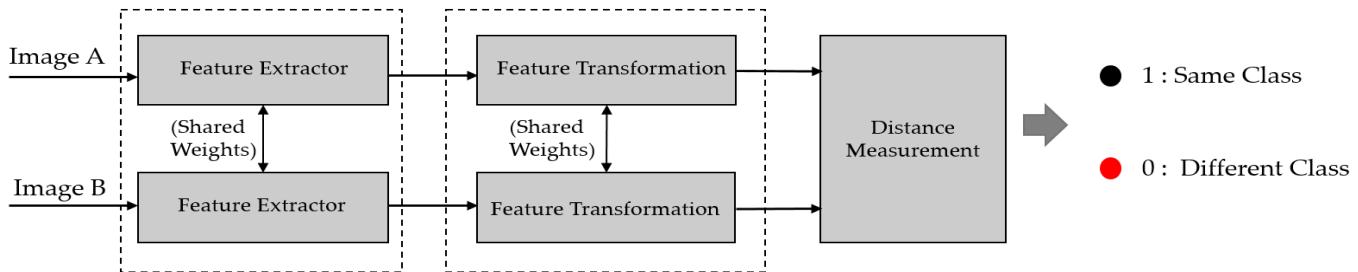
The feature maps are then manipulated by an operator  $C$  to further transform them into a feature space in which a better representation of the images can be obtained. Finally, the relation between two inputs ( $x_i$  and  $x_j$ ) is measured using a relation function  $f_r$  that eventually outputs a relation score ( $rsc$ ) between 0 and 1, where a score of 0 indicates that  $x_i$  and  $x_j$  are from different categories and a score of 1 indicates that the two samples are from the same category, as shown in Equation (11) [40].

$$rsc = f_r(C(f_e(x_i), f_e(x_j))) \quad (11)$$

Conventionally, a few-shot learning method works on three datasets: training, support, and testing. The training set is a large set that has a disjoint label space from the support and testing sets. The support set is a small set that shares the same label space as the testing set. We denote a few-shot learning method as  $n$ -way- $k$ -shot, where  $n$  indicates the number of classes and  $k$  indicates the number of samples in each class of the support set.

### 3.6.2. Proposed Few-Shot Learning-Based Network for Brain Tumor Image Classification

We designed a third classifier using the few-shot learning approach [40–42] as a similarity measurement network, as mentioned in Figure 1, and depicted in Figure 6. In this classifier, we designed a network that can distinguish the content of two images, that is, if the two images have a similar content, our network produces a high output value (close to 1); otherwise, our network produces a low output value (close to 0). We called this network a FS network in this study.



**Figure 6.** Design of our proposed FS network for measuring the distance between two input images.

As shown in Figure 6, our proposed network accepts two inputs (Images A and B in Figure 6). With these two images, we first used a pretrained CNN to extract the image feature vectors. The use of a pretrained CNN not only helps to extract efficient image features, but also removes the requirement for training an efficient network for image feature extraction. We further transformed the extracted image features into another feature space using an MLP network (Feature Transformation in Figure 6). The role of this MLP network is to convert image feature vectors from the original (extracted) space to another feature space in which the feature vectors of similar images are close together, while separating the feature vectors of two different images. The detail structure of MLP network is given in Table 3.

**Table 3.** Designing of the MLP-based feature transformation for the FS network.

Layer	#Neurons	Output Dimension	#Parameters
Input	0	$1 \times 5760$	0
Dense Layer 1	512	$1 \times 512$	2,949,632
Dense Layer 2	1024	$1 \times 1024$	525,312
Dense Layer 3	1	$1 \times 1$	1025
Total			3,475,969

To achieve this goal, we designed a distance measurement and loss function based on the cosine distance and binary cross-entropy. While we can use other types of distance measurements, such as Euclidian or Manhattan distance, the cosine distance has an ideal property in that it measures the content-based similarity between two images, instead of detailed position-based similarity as performed by Euclidian or Manhattan distance measurements. In addition, the cosine distance produces a distance measurement in the range of zero–one, in which an output of one indicates that the two images are completely similar, and an output of zero indicates two different images. The cosine distance is given by Equation (12). In this equation,  $x$  and  $y$  indicate the feature vectors extracted from two input images (Images A and B in Figure 6), and  $\theta$  represents the angle between these two feature vectors ( $x$  and  $y$ ).

$$\text{similarity} = \cos(\theta) = \frac{x \circ y}{|x||y|} \quad (12)$$

Finally, we used the binary cross-entropy function to evaluate the loss of training and update the weights of the FS network. As the ground-truth label of every image pair is

0 or 1, which indicates that the two images are different or similar, respectively, and the output of the network is also a cosine distance in the range of 0 to 1, we can easily calculate the loss between the ground-truth label and the output of the network using the binary cross-entropy function, as shown in Equation (13), where  $N$  is the batch size and  $y_i$  and  $o_i$  are the ground-truth label and output of the FS network, respectively.

$$\text{Loss} = -\frac{1}{N} \sum_{i=1}^N y_i \log(o_i) + (1 - y_i) \log(1 - o_i) \quad (13)$$

To classify an input image into predefined classes, we measure the cosine distance (the output of FS network mentioned in Figure 6) with all images in the support set. In an ideal case, if the input image is identical to one of the images in the support set, then the similarity score (cosine distance) is one. However, the new input image can never be same as images in support set. As a result, the output of FS network is a score value ranging from zero to one. By measuring the distance from the input image to all the support images, we classify the input image into predefined categories according to the best match (largest cosine distance) image in the support set.

### 3.7. Fusions of SVM, MLP, and FS

The last step in our proposed method is to combine the classification results of SVMs, MLP, and FS networks. For this purpose, we propose the use of score-level fusion methods based on SVMs, weighted-SUM, and weighted-PRODUCT rules. This is a popular fusion method and has been used by previous studies [43,44]. The output of each classifier (SVM, MLP, FS) is a tuple of non-binary numbers ranging from zero to one that indicates the probability of an input image sample belonging to a class. For example, we obtained the output probabilities for two classes as (0.1, 0.9) for an input image using the SVM classifier. This output result indicates that the input image has a probability of 10% belonging to the non-tumor class (the first class), whereas it has a higher probability of 90% belonging to the tumor class (the second class). For a classification system that uses a single classifier, the final decision is made by selecting the class with the highest probability, i.e., the input image belongs to the tumor class. For a classification system using multiple classifiers, we compared the SVM, weighted SUM and weighted PRODUCT rules to combine the output probabilities of individual classifiers.

For the first combination rule, we propose the use of the SVM-based method as the second-order classifier to combine the outputs of the first three classifiers: SVM, MLP, and FS. For this purpose, all outputs of the first three classifiers (SVM, MLP, and FS) are concatenated to form a new feature vector for training the second SVM layer.

For the second and third combination rules, we evaluated the classification performance by taking weights for each output of the first three classifiers (SVM, MLP, and FS) using the weighted-SUM (wSUM) and weighted-PRODUCT (wPROD), as shown in Equations (14) and (15), respectively. In these equations,  $\alpha_i$  indicates the weight value of the  $i$ th classifier (SVM, MLP, or FS) and  $S_{i,j}$  indicates the probability of the input image belonging to the class  $j$ th using the  $i$ th classifier; and the best weight values are determined experimentally to obtain the highest classification performance. Finally, the classification is performed by selecting the class that has the highest probability in a similar way as the classification using a single classifier. The optimal weight value of  $\alpha_i$  was experimentally determined with the training and validation data.

$$wSUM = \sum_{i=1}^N \alpha_i S_{i,j} \quad (14)$$

$$wPROD = \prod_{i=1}^N S_{i,j}^{\alpha_i} \quad (15)$$

For example, in the case of two classes, we obtained the output probabilities ( $S_{i,j}$  of Equation (14)) of two classes ( $j = 1$  and  $2$  of Equation (14)) by SVM ( $I = 1$  of Equation (14)), MLP ( $I = 2$  of Equation (14)), and FS classifiers ( $I = 3$  of Equation (14)) as (0.1, 0.9), (0.2, 0.8),

and  $(0.3, 0.7)$ , respectively. Using the weighted SUM rule with the weights ( $\alpha_i$  of Equation (14)) for the SVM, MLP, and FS classifiers of  $0.4, 0.5, 0.1$ , we obtained the final classification result as a tuple of  $(0.17 (=0.1 \times 0.4 + 0.2 \times 0.5 + 0.3 \times 0.1), 0.83 (=0.9 \times 0.4 + 0.8 \times 0.5 + 0.7 \times 0.1))$ . As a result, the final classification is made, so that the input image belongs to the second class ( $j = 2$  of Equation (14)) with a probability of  $0.83$  (83%). In our experiments, we measured the classification accuracy of all three combination rules using two public datasets and evaluated their performance in the brain tumor classification problem. The experimental results are presented in Section 4.

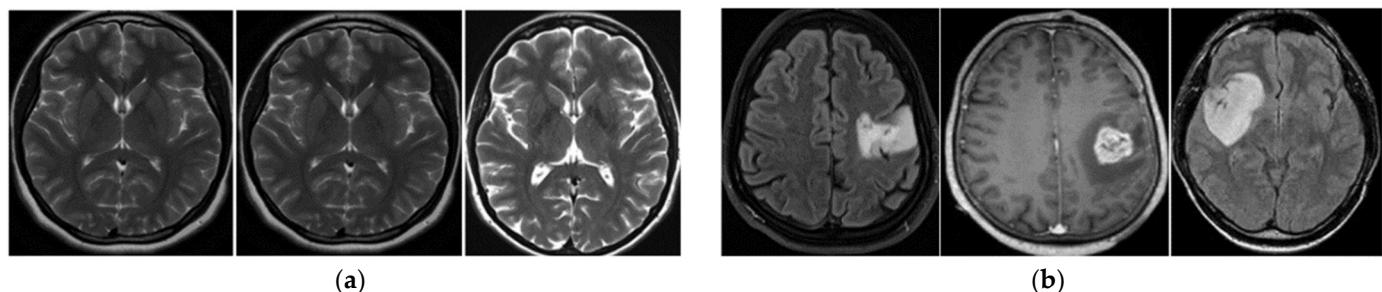
## 4. Experiments

### 4.1. Experimental Setups

We performed a five-fold cross-validation procedure to evaluate the performance of our proposed method using the BT-Small dataset. For this purpose, we randomly divided the entire BT-Small dataset into five equal parts and assigned them to the training, validation, and testing sets. However, in contrast to the conventional five-fold cross-validation procedure that uses most of the data for training and less data for testing, we used data from one part for training and validation and data from the remaining four parts for the testing dataset, as our study focuses on a small training dataset for the brain tumor image classification system. Among the images in one part that were assigned as the training and validation sets, we randomly selected an equal small number of images in each class for training, and the other images were used for validation. Based on this division, we used the training set to train the classifiers, which were then validated using the validation set. As the FS network mentioned in Section 3.6 requires a support set to evaluate the similarity function between a test image and the images in the support set, we randomly selected a subset of the testing set to form a support set for the FS network. The remaining data of the testing set were assigned to the final test set, which was used to measure the performance of the classifiers on real-world data. The experiments were repeated five times by iterating the training, validation, support and testing set division. Finally, the overall performance of the classification system was measured by taking the average performance of all five experiments. Table 4 summarizes the characteristics of the BT-Small dataset with detailed information on the training, validation, and testing sets. As can be observed from this table, the test dataset is much larger than the training and validation sets. In Figure 7, we show some examples of brain images from the BT-Small dataset.

**Table 4.** Summary of images in the BT-Small dataset.

Tumor Images			Non-Tumor Images			Total		
155			98			253		
Train	Validation	Test	Train	Validation	Test	Train	Validation	Test
Tiny-DB								
5	26	114	5	15	68	10	41	182
Small-DB								
10	21	114	10	10	68	20	31	182

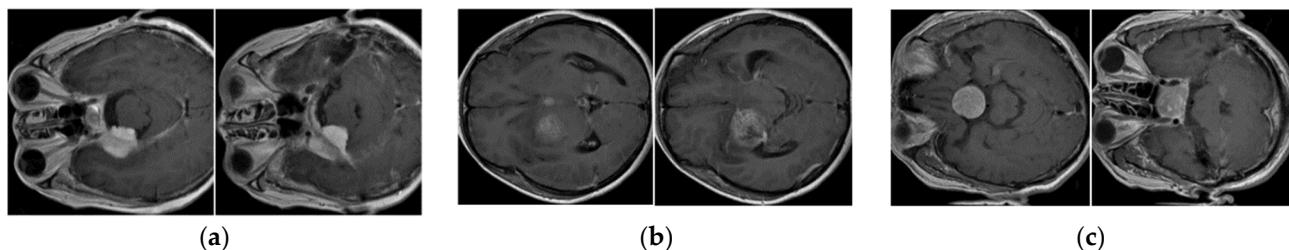


**Figure 7.** Example images in BT-Small: (a) Non-tumor brain images and (b) tumor brain images.

For the BT-Large dataset, the authors already divided this dataset into training (approximately 80% of the entire dataset) and testing parts (approximately 20% of the entire dataset). Similar to our experiments with the BT-Small dataset, we used a small part of the testing set for training and validation purposes, whereas the remaining part of the dataset was used for testing purposes, as we were dealing with the small training set problem. We repeated our experiments five times, and the final classification performance was measured by taking the average value of the performances obtained in each experiment. Table 5 provides a detailed description of the BT-Large dataset. Figure 8 shows some example images from this dataset.

**Table 5.** Summary of images in the BT-Large dataset.

Meningioma			Glioma			Pituitary			Total		
708			1426			930			3064		
Train	Validation	Test	Train	Validation	Test	Train	Validation	Test	Train	Validation	Test
Tiny-DB											
5	5	688	5	5	1406	5	5	910	15	15	3004
Small-DB											
10	10	678	10	10	1396	10	10	900	30	30	2974



**Figure 8.** Example of images in the BT-Large dataset: (a) Images of meningioma class; (b) images of glioma class; and (c) images of pituitary class.

To investigate the effects of the training data size, we performed two experiments using two training datasets with different numbers of training samples per class. In particular, we randomly selected images and formed two training datasets, namely a Tiny-DB that contains five images for each class and a Small-DB that contains ten images for each class, in our experiments with BT-Small and BT-Large datasets, as shown in detail in Tables 4 and 5. In addition, we randomly selected 10 images (10-shot) from each class to form a support set for classification using the FS network.

#### 4.2. Performance Measurement Metrics

To measure the performance of the classification system, we used the classification accuracy metric suggested in previous studies [23]. Suppose that a classification system

classifies an input image into  $N$  classes. Then, the classification accuracy is measured by Equation (16) as follows:

$$\text{Accuracy} = \frac{\sum_{i=1}^N \sum_j^M TC_{ij}}{\sum_{i=1}^N \sum_{j=1}^M TC_{ij} + \sum_{i=1}^N \sum_{j=1}^M FC_{ij}} \quad (16)$$

where  $TC_{ij}$  indicates the true classification of the  $j$ th images of the  $i$ th class (the image of the  $i$ th class was correctly classified as an image of the  $i$ th class), whereas  $FC_{ij}$  indicates the false classification of the  $j$ th images of the  $i$ th class (the image of the  $i$ th class was incorrectly classified as an image of another class). The values of  $N$  and  $M$  indicate the number of classes and images in each class, respectively. Consequently, classification accuracy indicates how well a classifier performs on a classification problem, and a high classification accuracy indicates better classification performance.

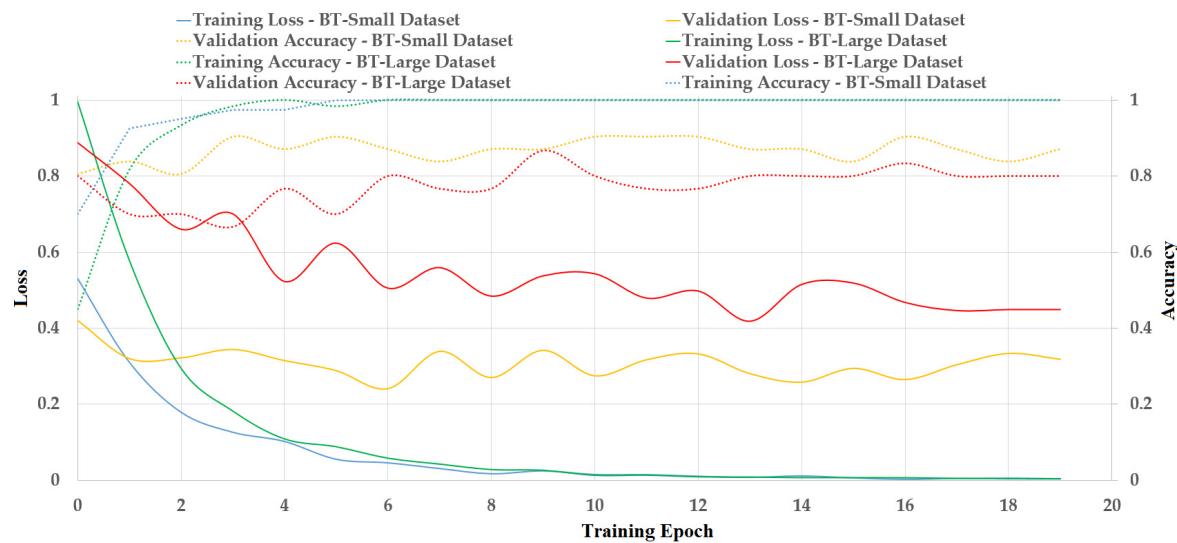
#### 4.3. Training of MLP and FS Network

In our proposed method, we use three classifiers, including SVM, MLP, and FS network. For the case of SVM, the image features were first extracted using pretrained CNN networks (Residual, Inception, and Dense network). With the extracted image features, we trained SVM model for classification using the conventional SVM method without the backpropagation procedure. For the cases of MLP and FS networks, we connected the outputs of the last convolution layers of pretrained CNN networks to the MLP layers to form a CNN-based or FS-based network and the weights of the MLP layers are trained using the backpropagation method.

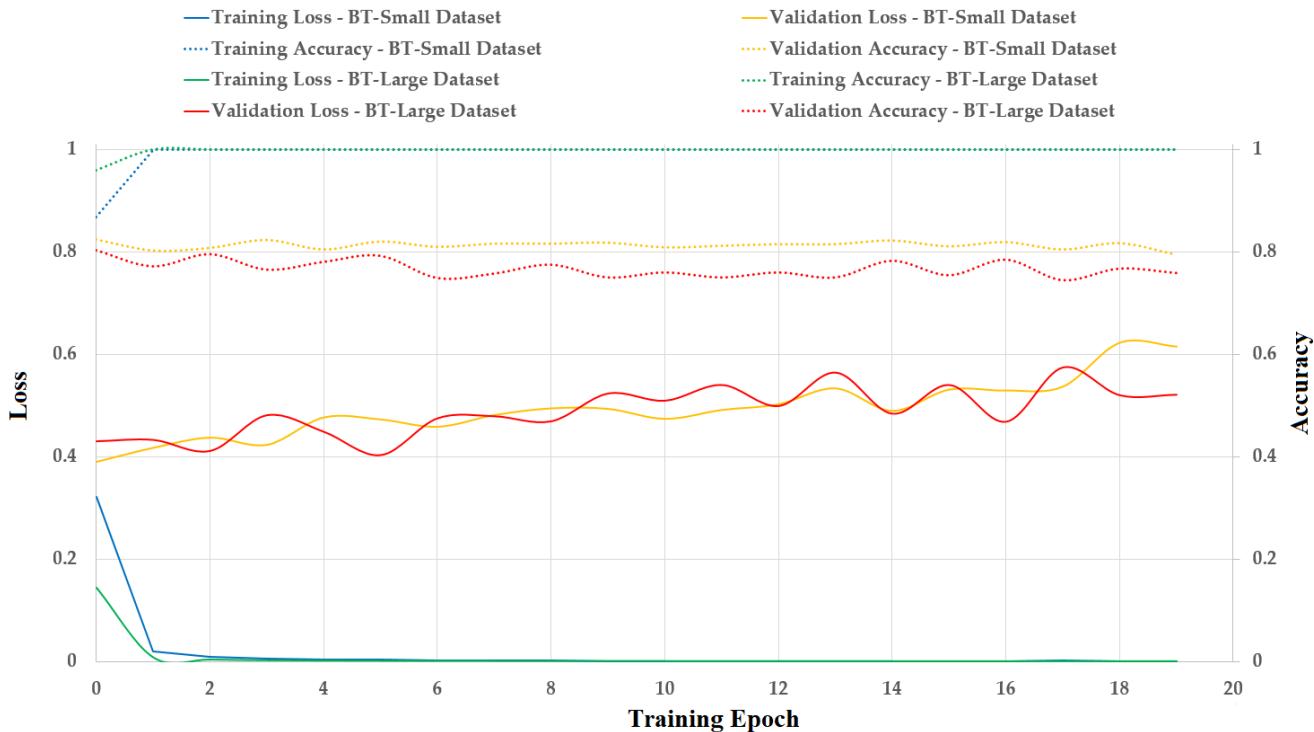
In our first experiment, we trained the MLP and FS network using the training datasets of the BT-Small and BT-Large datasets, as detailed in Section 4.1. In this experiment, we used the TensorFlow library [45] to implement and train the network. Table 6 lists the detailed parameters of the training procedures. As shown in this table, we used the Adam optimizer [46] to update the weights of these networks with a learning rate of 0.0001, and the network was trained for 20 epochs with a batch size of 32. Figures 9 and 10 show the results of our training procedure for the MLP and FS network, respectively. As shown in the Figures 9 and 10, we see that the loss and accuracy of the validation set do not approach to zero and 1, respectively, as those produced by the training set. As shown in these figures, the losses of the validation set are higher than those of the training set for the both cases of MLP and FS network. It is a common phenomenon as these networks are optimized for the training set, not validation set. It indicates some level of overfitting that caused by the fact that we are using very small data for training. However, the classification accuracies of the validation set are still high (more than 80% for the case of MLP as shown in Figure 9, and more than 76% for the case of FS as shown in Figure 10). Therefore, we can conclude that the MLP and FS network are not too overfitted with the training data and the training of the MLP and FS network is still successful.

**Table 6.** Parameters for the training procedure of the MLP and FS networks in our study.

Optimizer	Learning Rate	Epochs	Batch Size
Adam	0.0001	20	32



**Figure 9.** Training and validation curves (loss and accuracy) of the MLP network with the BT-Small and BT-Large datasets.



**Figure 10.** Training and validation curves (loss and accuracy) of the FS network with the BT-Small and BT-Large datasets.

#### 4.4. Evaluation of Classification Performance Using our Proposed Method in Comparison to Previous Studies

Based on the results of the training procedure, we further measured the classification performance by using the support and testing sets of the BT-Small and BT-Large datasets.

##### 4.4.1. Performance Evaluation on BT-Small Dataset

Table 7 shows the performance measurement of various classification methods, including the SVM-based and MLP-based methods, on the BT-Small dataset with two different sizes of the training dataset, that is, Tiny-DB (10 images in total, five images for each class) and Small-DB (20 images in total, 10 images for each class). As shown in this table, we

obtained the best classification accuracy of 75.903% using the MLP-based method with Tiny-DB. This classification value is higher than that obtained by the SVM-based method with four kernels: linear, radial basis function (RBF), polynomial, and sigmoid. When using Small-DB, the best classification accuracy using the MLP-based method was enhanced to 80.951%, whereas the classification accuracies of the SVM-based method also increased compared to those obtained when training with Tiny-DB, as shown in the last row of Table 7. These experimental results demonstrate that the SVM- and MLP-based methods are sufficient for brain tumor image classification using a small training dataset with the BT-Small dataset.

**Table 7.** Classification performance of SVM and MLP using the BT-Small dataset according to the size of the training data (unit: %).

Training Dataset	Single Classification Method					MLP	
	SVM-Based Method				Sigmoid		
	Linear	RBF	Poly				
Tiny-DB	72.626	69.437	71.857	74.050	75.903		
Small-DB	80.178	76.344	76.890	74.025	80.951		

Table 8 presents our experimental results for the cases using only the FS network and our proposed method, according to the size of the training set and number of shots. The upper part of this table presents the experimental results of our experiment with Tiny-DB with four shot values ( $k = 1, 3, 5$ , and 10). Specifically, we obtained the best classification accuracy for the FS network of 77.755%, which is higher than those obtained using the SVM- and MLP-based methods presented in Table 7. Using our proposed method, we obtained the best classification accuracy of 78.963% using the weighted SUM fusion rule.

**Table 8.** Classification performance of FS and the proposed method using BT-Small dataset according to the size of the training data and size of the support set (unit: %).

Training Dataset	FS		Proposed Method				
			SVM-Based Method				wSUM
	Shot	Accuracy	Linear	RBF	Poly	Sigmoid	wPROD
Tiny-DB	1	76.883	79.294	79.625	78.201	78.745	78.310
	3	76.882	76.328	75.667	76.443	78.524	77.653
	5	77.647	70.832	76.436	76.553	78.745	77.871
	10	77.755	77.756	75.228	77.103	78.305	78.963
Small-DB	1	82.039	82.148	81.273	80.060	81.491	83.033
	3	82.260	81.928	81.383	79.840	82.371	82.923
	5	82.260	82.037	81.383	79.949	82.371	82.923
	10	82.370	81.707	81.163	78.850	81.822	83.253

Similar results were obtained in our experiments with Small-DB, as shown in the lower part of Table 8. We obtained the best classification accuracy of 82.370% when using the FS network with a 10-shot setup. This result is also higher than those obtained using the SVM- and MLP-based methods, as presented in the last row of Table 7. Using the proposed method, we enhanced the classification results to 83.253% using the weighted SUM fusion rule. From these results, we confirmed that our proposed method is superior to all individual classification methods (SVM-, MLP-, and FS-based methods) when working with the BT-Small dataset.

#### 4.4.2. Performance Evaluation on BT-Large Dataset

Similar to Tables 7–10, show the experimental results obtained in our experiments with the BT-Large dataset according to the size of the training data and the various shot values for the FS network. Table 9 presents our experimental results for the cases of using SVM- and MLP-based methods using the BT-Large dataset with two different training set sizes, that is, a Tiny-DB (15 images in total, five images for each class) and a Small-DB (30 images in total, 10 images for each class). As shown in this table, we obtained the best classification accuracy of 48.995% when training the classification model using Tiny-DB and 49.159% when training the classification model using Small-DB using the SVM-based method with a polynomial kernel. These classification accuracies are higher than those produced by the MLP-based method (41.112% and 46.705% for the case of using the Tiny-DB and Small-DB datasets, respectively) and the other SVM kernels. However, we can see that all the classification accuracies in our experiments with the BT-Large dataset are lower than those with the BT-Small dataset. The reason for this is explained in more detail in Section 5.

**Table 9.** Classification performance of SVM- and MLP-based methods using the BT-Large dataset according to the size of the training data (unit: %).

Training Dataset	Single Classification Method					MLP	
	SVM-Based Method				Sigmoid		
	Linear	RBF	Poly				
Tiny-DB	48.189	41.771	48.995	42.064	41.112		
Small-DB	49.159	48.931	49.159	48.958	46.705		

**Table 10.** Classification performance of FS and the proposed method using the BT-Large dataset according to the size of the training data and size of the support set (unit: %).

Training Dataset	FS		Proposed Method				
	Shot	Accuracy	SVM-Based Method			wSUM	wPROD
			Linear	RBF	Poly		
Tiny-DB	1	41.085	49.527	44.794	44.308	45.786	50.685
	3	47.164	49.621	38.216	42.170	43.382	50.945
	5	48.882	49.441	38.315	41.891	43.263	51.238
	10	45.460	48.755	37.823	40.100	43.389	55.592
Small-DB	1	36.990	45.750	46.994	44.116	46.785	51.069
	3	40.161	45.387	46.557	41.607	45.683	51.197
	5	41.574	44.909	46.813	41.917	46.147	51.177
	10	45.602	42.811	46.806	43.067	44.950	51.446

Table 10 shows the experimental results for the FS network and the proposed method. We obtained the best classification accuracy of 48.882% for the FS network when using Tiny-DB and a classification accuracy of 45.602% when using Small-DB. Using our proposed method, we obtained the best classification accuracy of 55.792% using the fusion rule based on the weighted PRODUCT with Tiny-DB and a classification accuracy of 51.708% using Small-DB. These classification accuracies are higher than those produced by the individual classification methods. This result again confirms that our proposed method is more efficient than the individual classification method for brain tumor classification problems with the BT-Large dataset. However, we observed from Tables 9 and 10 that the classification accuracies obtained with the BT-Large dataset were low for all the cases

of the classification method (SVM, MLP, FS, and our proposed method). These results demonstrate the strong negative effects of the size of the training data on classification accuracy using the BT-Large dataset. The reason for this low classification accuracy is discussed in more detail in Section 5.

#### 4.4.3. Performance Evaluation Using Single Network for Feature Extraction

As shown in Figure 1, our proposed method used three pretrained CNN networks to extract image features. While this approach takes long processing and makes the classification system become complex, it was used in order to extract rich information from input images. To validate the efficiency of this design, we performed an additional experiments using a less complex system in which the three pretrained CNN networks are replaced by a simple and less complex pretrained CNN network. For this purpose, we used the MobileNet network [47] instead of the combination of three networks (Residual, Inception, and Dense network) mentioned in Figure 1. The MobileNet network was introduced by Google research team that used the depthwise separable convolution to reduce the complexity and number of network parameters. As a result, the MobileNet network is a light network that is suitable for mobile applications. We called this setup a MobileNet-based classification system for convenience. The detailed experimental results are presented in Table 11.

**Table 11.** Comparison of the classification accuracies between the MobileNet-based classification system and our proposed method (unit: %).

Dataset	MobileNet-Based System	Proposed System
BT-Small	83.133	83.253
BT-Large	43.782	55.792

As observed from Table 11, the MobileNet-based classification system yielded a classification accuracy of 83.133% using the BT-Small dataset, and an accuracy of 43.782% using the BT-Large dataset. As mentioned in Tables 8 and 10, we obtained the best classification accuracies of 83.253% and 55.792% for the cases of using the BT-Small and BT-Large datasets, respectively. We see that the classification accuracies are similar between our proposed method and the MobileNet-based classification method for the case of using the BT-Small dataset (83.133% vs. 83.353%). However, our proposed method outperformed the MobileNet-based classification method when working with the BT-Large dataset (43.782% vs. 57.792%). Through these experimental results, we conclude that the use of multiple pretrained CNN networks for image feature extraction is efficient for extracting richer information from input images than the use of a single and less complex pretrained CNN network.

#### 4.4.4. Performance Evaluation Using F1Score Metric

In a multi-class classification system, the classification accuracy, as presented in Equation (16), has been widely used. In a medical image classification system, the classification accuracy is the measurement of a classification system ability to detect both the positive and negative cases. To investigate the classification performance in detecting positive cases (i.e., detecting disease), we further used the F1Score metric to measure the performance of our proposed method. By definition, the F1Score is a harmonic mean of two metrics: *precision* and *recall*, as shown in Equation (19). In detail, the precision is the measurement of a classification system ability to detect the positive cases (*TP*) and reduce the errors of incorrectly recognizing a negative case as a positive case (false positive—*FP*) as shown in Equation (17); whereas the *recall* is the measurement of a classification system ability to detect the positive cases and reduce the errors of incorrectly rejecting a positive case (false negative—*FN*), as shown in Equation (18). In these equations, *K* indicates the number of classes. Therefore, the *precision* and *recall* metrics are normally used to evaluate the ability of

a classification system in detecting disease. As observed from Equations (17) and (18), the higher values of precision and recall indicate high performance of a classification system on detecting disease. However, there is a trade-off between *precision* and *recall*. Therefore, the F1Score, as shown in Equation (19), is normally used to measure the performance of a medical image classification system.

$$Precision = \frac{1}{K} \sum_{i=1}^K \frac{TP_i}{TP_i + FP_i} \quad (17)$$

$$Recall = \frac{1}{K} \sum_{i=1}^K \frac{TP_i}{TP_i + FN_i} \quad (18)$$

$$F1Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (19)$$

The detailed experimental results using the F1Score are given in Table 12. As shown in this table, our proposed method outperformed all of these individual classification methods by producing the highest F1Score values of 82.330% and 51.888% for the cases of using the BT-Small and BT-Large datasets, respectively. From the experimental results, we conclude that our proposed method is efficient for recognizing the positive cases and outperforms the SVM-, MLP-, and FS-based classification methods.

**Table 12.** Evaluation of the classification performance (F1Score) of our proposed method in comparison to the SVM-, MLP-, and FS-based methods (unit: %).

Dataset	SVM-Based Method				MLP-Based Method	FS-Based Method	Proposed Method
	Linear	RBF	Poly	Sigmoid			
BT-Small	79.608	75.78	76.518	73.897	80.182	80.898	82.330
BT-Large	44.843	35.192	45.136	37.027	39.005	43.491	51.888

#### 4.4.5. Comparison with Previous Studies

As the final experiment, we compared the classification performance of our proposed method with those produced by previous studies. For this purpose, we compared the performance of our proposed method with two previous classification approaches, including the SVM-based and fine-tuning CNN-based method. In the first approach, Kang et al. [23] successfully used three pretrained networks (Residual, Inception, and Dense network) to extract image features and used the SVM for classifying the brain image into predefined categories. For the second approach, fine-tuning of a CNN network was used for the classification problem [16–18]. These approaches are suitable for working with a small training dataset because the pretrained CNN networks are suitable for extracting efficient image features without a training procedure [23]. The first approach was used in our work, as described in Section 2, and the experimental results were reported in Tables 7 and 9 for the cases of using BT-Small and BT-Large, respectively.

For the second approach, we performed various experiments by fine-tuning four different CNN network architectures, including the residual, inception, dense, and mobile network. For each CNN network architecture, we reused all the convolution layers, and replaced the fully connected layers with the new ones according to the number of output nodes (i.e., 2 with the BT-Small dataset, and 3 with the BT-Large dataset). We trained and evaluated these networks using the same data with our experiment for measuring the performance of our proposed method, as mentioned in Sections 4.4.1–4.4.3. The final experimental results are given in Table 13. As shown in this table, our proposed method outperformed all of the previous studies (SVM-based, and fine-tuning-based methods) using both the BT-Small and BT-Large dataset. In detail, we obtained accuracy of roughly 3% (83.253% vs. 80.178%) better than the SVM-based method; and more than 10% (83.253%

vs. 72.504%) better than the fine-tuning-based method using the BT-Small dataset. With the BT-Large dataset, we obtained an accuracy of roughly 6% (55.792% vs. 49.159%) better than the SVM-based method; and roughly 5.5% (55.792% vs. 50.108%) better than the fine-tuning-based method. Through these experimental results, we conclude that our proposed method is efficient for enhancing the classification performance of previous studies when working with the small training dataset.

**Table 13.** Comparison on the classification performance (accuracy and F1Score) between the proposed method and previous studies (unit: %).

Dataset	Metrics	SVM-Based Method [23]	DenseNet-Based Method	InceptionNet-Based Method [18]	ResNet-Based Method [17]	MobileNet-Based Method	Proposed Method
BT-Small	Accuracy	80.178	70.632	72.504	57.475	62.642	83.253
	F1Score	79.608	68.779	71.196	55.779	60.682	82.330
BT-Large	Accuracy	49.159	50.108	46.490	34.089	44.405	55.792
	F1Score	45.136	47.795	44.950	32.249	41.249	51.888

## 5. Discussion

A large dataset for brain tumor segmentation purpose exists, namely the BraTS dataset [48–50]. This dataset contains 3D MRI scans of human brain with tumors. Therefore, it does not fit with our study as we tend to classify 2D MRI images. In addition, although we can use tumor-free slices as non-tumor and the rest as tumor images, it makes difficulties for training and testing of our proposed method because slices in 3D MRI scans exhibit high correlation that makes it difficult to train classification methods (SVM, MLP, and FS). Therefore, we did not use the BraTS dataset in our experiments. Instead, we used two public datasets that have been used in previous study [23] to evaluate the performance of our proposed method, including the BT-small [28] and BT-large datasets [29]. These datasets were used in a previous study for the brain tumor image classification problem [15,16,18,20,23] and open for all researchers. Therefore, the use of these datasets allow other researches to easily use and compare with our study.

In a previous study conducted by Kang et al. [23], a very high classification accuracy was obtained using an SVM- or CNN-based method, approximately 98.04% with the BT-Small dataset, and 93.72% with the BT-Large dataset. Using the BT-Large dataset, previous studies conducted by Abiwinanda et al. [5], Swati et al. [16], Deepak et al. [18], Bodapati et al. [20], Díaz-Pernas et al. [22], and Isunuri et al. [15] showed very high classification accuracies, ranging from 84% to more than 98% when training with a large amount of images. From our experimental results, we confirmed that the lack of training data is a critical problem in a classification system as lower classification accuracies were produced by our experiments compared to those produced by previous studies using the same dataset. This is because previous studies performed training on a large dataset and testing on a small dataset. However, our study focuses on a different scenario, in which the classification system lacks training data. In this special case, the performance of conventional methods, such as SVM- or CNN-based methods (features extraction by CNN followed by MLP for classification), is significantly reduced, as shown in our experimental results presented in Tables 7 and 9. However, as shown in Tables 8 and 10, our proposed method enhanced the overall classification accuracy by combining the results of the SVM- and MLP-based methods with a new classification method based on the few-shot learning technique. Based on these experimental results, we believe that our proposed method is more efficient than previous methods and helps to enhance the classification accuracy for classification problems that lack training data.

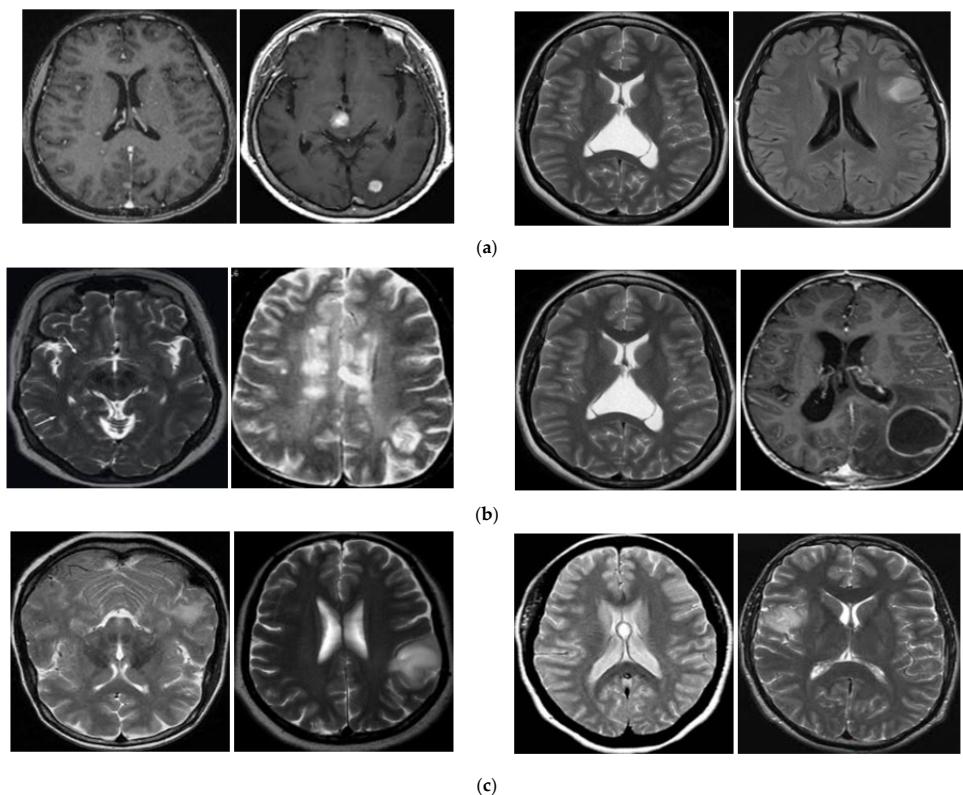
As explained in Section 3.1, the BT-Large dataset contains brain images that belong to three predefined categories: Meningioma, glioma, and pituitary, as shown through some examples in Figure 8. This dataset is different from the BT-Small dataset, which contains images of two categories of tumor (with tumor region) and non-tumor (without

any tumor region) images. As the BT-Small dataset contains images with and without tumors, the difference between the images in the two categories in the BT-Small dataset is high. Therefore, we can easily distinguish between tumor and non-tumor images. However, the difference between images among categories in the BT-Large dataset was low because all images in the BT-Large dataset contained tumors. In addition, the BT-Small dataset contained two image classes (categories), whereas the BT-Large dataset contained three image classes (meningioma, glioma, and pituitary). Therefore, classifying images into a large number of predefined categories using a small amount of training data is also a more difficult problem than classifying images into a small number of predefined categories. Therefore, the classification performance of the BT-Small dataset was much better than that of the BT-Large dataset, as shown in Tables 7–10. For a 3-class classification system, a random guessing system produces an accuracy of about 33.333% that is much lower than our classification accuracy (55.792% mentioned in Table 10). In addition, Table 13 shows that our proposed method also outperformed various classification methods when using small data for training. From this result, we think that our algorithm successfully enhanced the classification performance compared to previous classification methods. From our experimental results, we showed that the lack of training data lead to reduction in performance of conventional classification systems. Knowing this problem, our proposed method can be used in real-world applications when there is less data for training the classification system such as a new or less common disease is presented.

Classifying images into a large number of predefined categories is more difficult a problem than classifying images into a smaller number of categories. This problem is caused by the fact that there always exist the inter-class and intra-class correlation among images in the dataset. As a result, when the number of classes (predefined categories) increases, it causes more errors because of the interclass correlation. To reduce the negative effects of this problem, a possible solution is that we should collect more data and perform data augmentation to make the dataset generalize for a general classification problem. In addition, we could perform a pre-classification scheme that classifies the data into some major categories before fine-tuning into detailed categories.

In Figure 11, we show some example error cases caused by individual method, including the SVM-, MLP-, and FS-based methods. As shown in this figure, all these methods (SVM-, MLP-, and FS-based methods) can falsely classify a non-tumor image as a tumor image when the input images contain noise and/or high-contrast brain regions, as shown in the images on the left side of this figure. When the size of the tumor is relatively small or unclear, these methods can falsely classify a tumor image as a non-tumor image, as shown in the images on the right side of this figure.

As shown in Section 4, our proposed method outperformed previous methods for brain tumor image classification using a small training dataset. For demonstration purposes, we show some examples in which our proposed method (based on the weighted-SUM rule) helps to enhance the classification performance of individual classifiers in Figure 12. We can observe from this figure that, although an individual method (among three methods) can falsely classify an image, the fusion with the result of other classifiers helps to correct that error and consequently enhances the classification performance of the overall system.



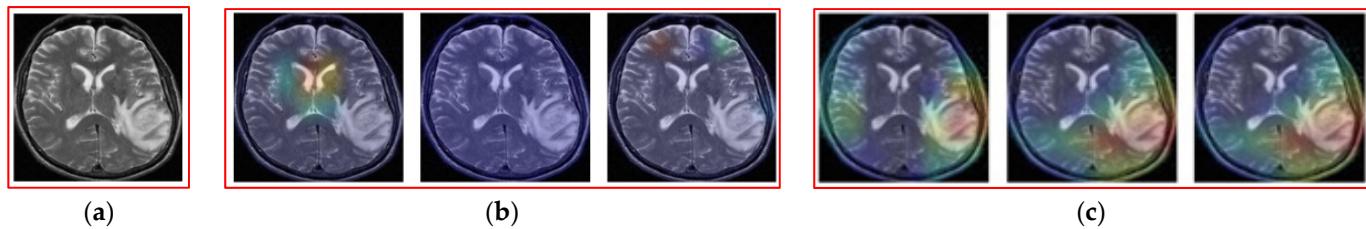
**Figure 11.** Examples of error cases of single classification methods: (a) SVM-based method, (b) MLP-based method, and (c) FS-based method. (Left pair: Non-tumor to tumor cases; Right pair: Tumor to non-tumor cases).

	<ul style="list-style-type: none"> <li>▪ Ground-truth Label : 1</li> <li>▪ SVM-based Method : 0</li> <li>▪ MLP-based Method : 1</li> <li>▪ FS-based Method : 1</li> <li>▪ Proposed Method : 1</li> </ul>		<ul style="list-style-type: none"> <li>▪ Ground-truth Label : 1</li> <li>▪ SVM-based Method : 1</li> <li>▪ MLP-based Method : 0</li> <li>▪ FS-based Method : 1</li> <li>▪ Proposed Method : 1</li> </ul>
	<ul style="list-style-type: none"> <li>▪ Ground-truth Label : 0</li> <li>▪ SVM-based Method : 0</li> <li>▪ MLP-based Method : 0</li> <li>▪ FS-based Method : 1</li> <li>▪ Proposed Method : 0</li> </ul>		<ul style="list-style-type: none"> <li>▪ Ground-truth Label : 0</li> <li>▪ SVM-based Method : 1</li> <li>▪ MLP-based Method : 0</li> <li>▪ FS-based Method : 0</li> <li>▪ Proposed Method : 0</li> </ul>

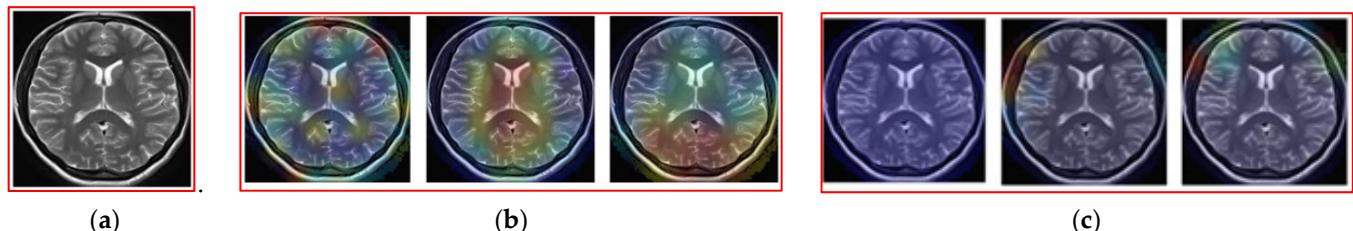
**Figure 12.** Example of enhancement cases using our proposed method (Label 0 indicates the non-tumor image; Label 1 indicates the tumor image).

To investigate the internal functioning of the MLP-based method for the classification problem, we extracted the class activation map (CAM) image [51] of example input images using a CNN-based network (feature extracted by CNN and followed by classification by MLP) presented in Figure 4, as shown in Figures 13 and 14, for the case where the input image is a tumor image (Figure 13) and for the case where the input image is a non-tumor image (Figure 14). As shown in Figure 13, the MLP network pays more attention to the tumor region (higher weight) than the other regions for making a decision on the input

image belonging to the tumor class (Figure 13c), whereas it uniformly analyzes the overall input image for making a decision on the input image belonging to the non-tumor class (Figure 13b). When the input image is a non-tumor image, as shown in Figure 14, the networks pay attention to the entire brain region for making decisions on the input image belonging to the non-tumor class, whereas they pay attention to some high-frequency regions such as the skull or brain boundary region to make decisions on the input image belonging to a tumor class. From these figures, we see that the MLP network pays attention to tumor regions or high-frequency regions for making decisions on input images belonging to the tumor region and pays uniform attention to the entire brain region to make decisions on input images belonging to the non-tumor class.



**Figure 13.** Examples of CAM images using MLP network: (a) Input image with tumor lesion (bright region on image), (b) CAM image at the last convolution layer of the DenseNet (Left), InceptionNet (Middle), and ResidualNet (Right) with respect to the non-tumor class; and (c) CAM images at the last convolution layer of the DenseNet (Left), InceptionNet (Middle), and ResidualNet (Right) with respect to the tumor class.



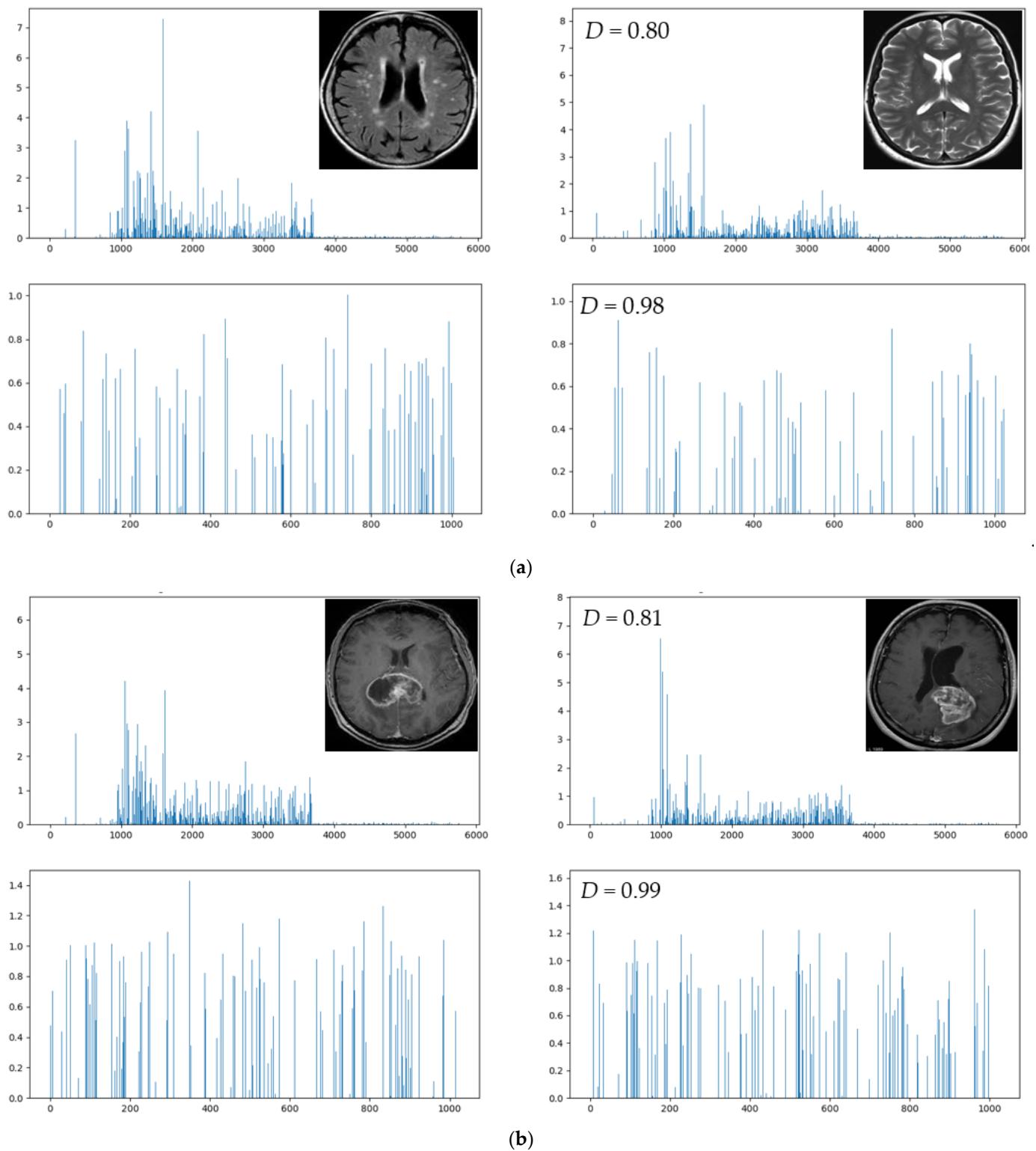
**Figure 14.** Examples of CAM images using MLP network: (a) Input image without tumor region, (b) CAM image at the last convolution layer of the DenseNet (Left), InceptionNet (Middle), and ResidualNet (Right) with respect to the non-tumor class; and (c) CAM images at the last convolution layer of the DenseNet (Left), InceptionNet (Middle), and ResidualNet (Right) with respect to the tumor class.

To investigate the internal functioning of the FS network, we plotted the extracted image features before and after transformation using the FS network of some input image pairs, as shown in Figures 15 and 16. In these figures, the  $D$  value indicates the measured cosine distance between two extracted image features. The upper part of Figure 15a shows the original deep image features extracted by pretrained CNNs, and the cosine distance between the two features is approximately 0.8 ( $D = 0.80$  in Figure 15). This indicates that the two images are quite similar (close to 1). By transforming the input image features to other spaces using the FS network, as shown in the lower part of Figure 15a, the cosine distance between two input images is increased to roughly 0.98 ( $D = 0.98$  in Figure 15a), which indicates that the two images are very close together. This is what we expect, as they are all in the same class of non-tumor images. Similarly, Figure 15b shows the case of two tumor input images. Without the use of the FS network, the similarity of two input images is approximately 0.81 ( $D = 0.81$  in Figure 15b). Using our proposed FS network, the similarity becomes 0.99 ( $D = 0.99$  in Figure 15b), which indicates that the two images belong to the same category.

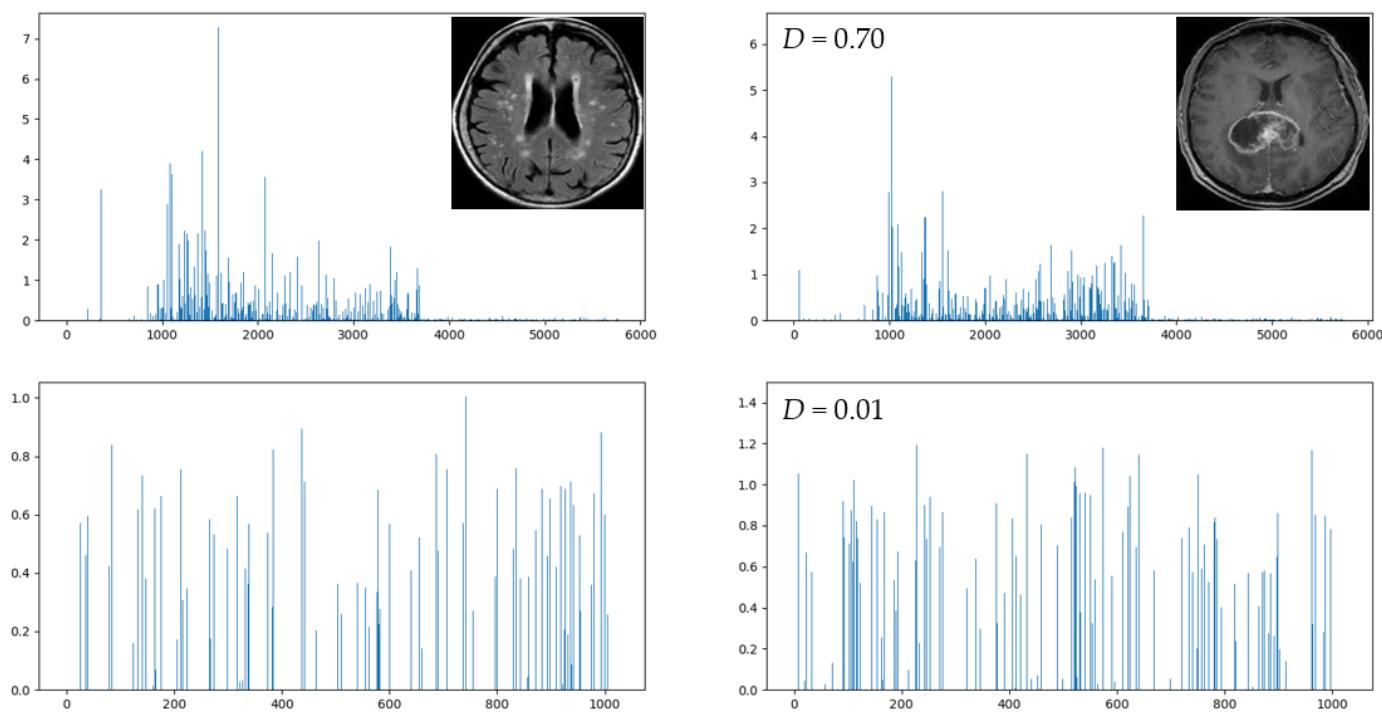
Figure 16 shows a case in which two input images were from different classes. As shown in the upper part of this figure, the cosine distance between the two images is

obtained as 0.70 ( $D = 0.70$  in Figure 16) using the deep features extracted using pretrained CNNs. This value indicates that the contents of the two images are quite similar, even though they belong to two different classes. Using our proposed FS network, the distance between two input images is reduced to 0.01 ( $D = 0.01$  in Figure 16), which indicates that the two images are different (cosine distance is now close to 0). This is what we expect because the two images are from two different classes. From Figures 15 and 16, we can see that the FS network helps to adjust the similarity measurement when two images from the same class will have high similarity measurements, whereas the two images from different classes will have very low similarity measurements.

In our experiments, we used less data for training, and a large amount of data for testing to simulate the situation in which a new or less common disease appears, as explained in Section 4. The use of less data for training is more difficult than training deep learning models using a large amount of data because of the lack of information. Due to this problem, some level of overfitting in the classification models exists, which makes the testing result worse than the training result. As a result, the classification result decreases when more data are used for testing. As shown in Tables 4 and 5, the number of training images are similar in both cases of BT-Small and BT-Large, but the number of testing images in BT-Large is much larger than that in BT-Small.



**Figure 15.** Example of image features extracted using pretrained CNNs (upper rows) and using our proposed FS network (lower rows) in the case of two images in the same class: (a) Two different images in the non-tumor class and (b) two different images in the tumor class.



**Figure 16.** Examples of image features extracted using pretrained CNNs (upper rows) and using our proposed FS network (lower rows) in the case of two images from different classes (left: non-tumor brain image; right: tumor brain image).

As shown in Tables 8, 10 and 13, our study produced higher performance than those of previous studies when working with small training data. While the errors produced by our proposed method are high (17% for the binary problem, and 44% for the ternary problem), they are much lower than that of a random classification system. In addition, we are working with a computer-aided diagnosis (CAD) system. Therefore, the results produced by our proposed method cannot be used directly in the diagnosing and treatment process. Instead, our proposed method provides suggestions for doctors in a double screening procedure. Therefore, it can be used to help enhance the disease diagnosis and treatment performance conducted by doctors.

BT-Small and BT-large datasets used in our experiments were public datasets which have been widely used for previous research. They have the image variations including some level of the mix of orientations and contrast changes. As these datasets were originally collected considering the MRI images of real application, we used these datasets including the image variations for our experiments.

## 6. Conclusions

This paper deals with the image-based classification problem when less data is available for training. For this purpose, we proposed a classification method based on the score level fusion of multiple classification methods. In detail, we used three classifiers of SVM, MLP, and few-shot learning to classify input images into several categories of brain diseases. As each method has advantages and disadvantages in classification problems, we used the score level fusion technique based on SVM, weighted-SUM, and weighted-PRODUCT rules to enhance the classification performance of individual methods. We applied our proposed method on the brain tumor image classification problem and showed that our proposed method outperformed individual methods (SVM, MLP, and few-shot) as well as the fine-tuning of various CNN networks (Residual, Inception, Dense, and Mobile network) using two public brain tumor image datasets. In addition, we observed from our experimental results that the classification performance was still low for all classification methods. Therefore, we plan to investigate this issue in future research, such as using generative

models to learn and generate additional images for training, and using pretrained models that are trained using MRI images for image feature extraction; investigate the effects of the depth of MLP network on the classification system; and using less complex mixture in the dataset (contrast and/or orientations).

**Author Contributions:** Methodology, D.T.N.; validation, S.H.N., G.B. and M.O.; supervision, K.R.P.; writing—original draft, D.T.N.; writing—review and editing, K.R.P. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was supported in part by the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT (MSIT) through the Basic Science Research Program (NRF-2020R1A2C1006179), in part by the NRF funded by the MSIT through the Basic Science Research Program (NRF-2021R1F1A1045587), and in part by the NRF funded by the MSIT through the Basic Science Research Program (NRF-2022R1F1A1064291).

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Castiglione, A.; Vijayakumar, P.; Nappi, M.; Sadiq, S.; Umer, M. COVID-19: Automatic detection of the novel coronavirus disease from CT images using an optimized convolutional neural network. *IEEE Trans. Ind. Inform.* **2021**, *17*, 6480–6488. [[CrossRef](#)]
2. Vasilakakis, M.; Iosifidou, V.; Fragkaki, P.; Iakovidis, D. Bone fracture identification in X-ray images using fuzzy wavelet features. In Proceedings of the 19th IEEE International Conference on Bioinformatics and Bioengineering (BIBE), Athens, Greece, 28–30 October 2019; pp. 726–730. [[CrossRef](#)]
3. Nguyen, D.T.; Pham, T.D.; Batchuluun, G.; Yoon, H.; Park, K.R. Artificial intelligence-based thyroid nodule classification using information from spatial and frequency domains. *J. Clin. Med.* **2019**, *8*, 1976. [[CrossRef](#)]
4. Ibrahim, W.H.; Osman, A.; Mohamed, Y.I. MRI brain image classification using neural networks. In Proceedings of the International Conference on Computing, Electrical and Electronics Engineering, Khartoum, Sudan, 26–28 August 2013; pp. 253–258. [[CrossRef](#)]
5. Abiwinanda, N.; Hanif, M.; Hesaputra, S.T.; Handayani, A.; Mengko, T.R. Brain tumor classification using convolutional neural network. In *World Congress on Medical Physics and Biomedical Engineering 2018*; Springer: Gateway East, Singapore, 2019; pp. 183–189. [[CrossRef](#)]
6. Yahiaoui, A.F.Z.; Bessaïd, A. Segmentation of ischemic stroke area from CT brain images. In Proceedings of the International Symposium on Signal, Image, Video and Communications (ISIVC), Tunis, Tunisia, 21–23 November 2016; pp. 13–17. [[CrossRef](#)]
7. Burgos, N.; Bottani, S.; Faouzi, J.; Thibeau-Sutre, E.; Colliot, O. Deep learning for brain disorders: From data processing to disease treatment. *Brief. Bioinform.* **2021**, *22*, 1560–1576. [[CrossRef](#)]
8. Cheng, J.; Yang, W.; Huang, M.; Huang, W.; Jiang, J.; Zhou, Y.; Yang, R.; Zhao, J.; Feng, Y.; Feng, Q.; et al. Retrieval of Brain Tumors by Adaptive Spatial Pooling and Fisher Vector Representation. *PLoS ONE* **2016**, *11*, e0157112. [[CrossRef](#)]
9. Huang, M.; Yang, W.; Yu, M.; Lu, Z.; Feng, Q.; Chen, W. Retrieval of Brain Tumors with Region-Specific Bag-of-Visual-Words Representation in Contrast-Enhanced MRI Images. *Comput. Math. Methods Med.* **2012**, *2012*, 280538. [[CrossRef](#)]
10. Ismael, M.R.; Abdel-Qader, I. Brain tumor classification via Statistical Features and Back-propagation Neural Network. In Proceedings of the IEEE International Conference on Electro/Information Technology (EIT), Rochester, MI, USA, 3–5 May 2018; pp. 0252–0257. [[CrossRef](#)]
11. Gurbina, M.; Lascu, M.; Lascu, D. Tumor Detection and Classification of MRI Brain Image using Different Wavelet Transforms and Support Vector Machines. In Proceedings of the 42nd International Conference on Telecommunications and Signal Processing (TSP), Budapest, Hungary, 1–3 July 2019; pp. 505–508. [[CrossRef](#)]
12. Zaw, H.T.; Maneerat, N.; Win, K.Y. Brain tumor detection based on Naïve Bayes Classification. In Proceedings of the 5th International Conference on Engineering, Applied Sciences and Technology (ICEAST), Luang Prabang, Laos, 2–5 July 2019; pp. 1–4. [[CrossRef](#)]
13. Minz, A.; Mahobiya, C. MR Image Classification Using Adaboost for Brain Tumor Type. In Proceedings of the 7th International Advance Computing Conference (IACC), Hyderabad, India, 5–7 January 2017; pp. 701–705. [[CrossRef](#)]
14. Chen, Y.; Yin, M.; Li, Y.; Cai, Q. CSU-Net: A CNN-Transformer parallel network for multimodal brain tumour segmentation. *Electronics* **2022**, *11*, 2226. [[CrossRef](#)]
15. Isunuri, B.V.; Kakarla, J. Three-class brain tumor classification from magnetic resonance images using separable convolution based neural network. *Concurr. Comput. Pr. Exp.* **2022**, *34*, e6541. [[CrossRef](#)]
16. Swati, Z.N.K.; Zhao, Q.; Kabir, M.; Ali, F.; Ali, Z.; Ahmed, S.; Lu, J. Brain tumor classification for MR images using transfer learning and fine-tuning. *Comput. Med. Imaging Graph.* **2019**, *75*, 34–46. [[CrossRef](#)]
17. Kumar, R.L.; Kakarla, J.; Isunuri, B.V.; Singh, M. Multi-class brain tumor classification using residual network and global average pooling. *Multimed. Tools Appl.* **2021**, *80*, 13429–13438. [[CrossRef](#)]

18. Deepak, S.; Ameer, P.M. Brain tumor classification using deep CNN features via transfer learning. *Comput. Biol. Med.* **2019**, *111*, 103345. [[CrossRef](#)]
19. Alanazi, M.F.; Ali, M.U.; Hussain, S.J.; Zafar, A.; Mohatram, M.; Irfan, M.; AlRuwaili, R.; Alruwaili, M.; Ali, N.H.; Albarrak, A.M. Brain Tumor/Mass Classification Framework Using Magnetic-Resonance-Imaging-Based Isolated and Developed Transfer Deep-Learning Model. *Sensors* **2022**, *22*, 372. [[CrossRef](#)]
20. Bodapati, J.D.; Shaik, N.S.; Naralasetti, V.; Mundukur, N.B. Joint training of two-channel deep neural network for brain tumor classification. *Signal Image Video Process.* **2021**, *15*, 753–760. [[CrossRef](#)]
21. Toğacıar, M.; Ergen, B.; Cömert, Z. Tumor type detection in brain MR images of the deep model developed using hypercolumn technique, attention modules, and residual blocks. *Med. Biol. Eng. Comput.* **2021**, *59*, 57–70. [[CrossRef](#)]
22. Díaz-Pernas, F.J.; Martínez-Zarzuela, M.; Antón-Rodríguez, M.; González-Ortega, D. A deep learning approach for brain tumor classification and segmentation using a multiscale convolutional neural network. *Healthcare* **2021**, *9*, 153. [[CrossRef](#)]
23. Kang, J.; Ullah, Z.; Gwak, J. MRI-Based brain tumor classification using ensemble of deep features and machine learning classifiers. *Sensors* **2021**, *21*, 2222. [[CrossRef](#)]
24. Deepak, S.; Ameer, P.M. Automated categorization of brain tumor from MRI using CNN features and SVM. *J. Ambient. Intell. Humaniz. Comput.* **2021**, *12*, 8357–8369. [[CrossRef](#)]
25. Kesav, N.; Jibukumar, M.G. Efficient and low complex architecture for detection and classification of brain tumor using RCNN with two channel CNN. *J. King Saud Univ.—Comput. Inf. Sci.* **2022**, *34*, 6229–6242. [[CrossRef](#)]
26. Chatterjee, S.; Nizamani, F.A.; Nurnberger, A.; Speck, O. Classification of brain tumours in MR images using deep spatiotemporal models. *Sci Rep.* **2022**, *12*, 1505. [[CrossRef](#)]
27. Github Site for BrainTumorNet. 2022. Available online: <https://github.com/idahousa/BrainTumorClassification> (accessed on 1 October 2022).
28. Chakrabarty, N. Brain MRI Images for Brain Tumor Detection Dataset. Available online: <https://www.kaggle.com/datasets/navoneel/brain-mri-images-for-brain-tumor-detection> (accessed on 1 October 2022).
29. Cheng, J. Brain Tumor Dataset. Available online: [https://figshare.com/articles/dataset/brain\\_tumor\\_dataset/1512427?file=7953\\_679](https://figshare.com/articles/dataset/brain_tumor_dataset/1512427?file=7953_679) (accessed on 1 October 2022).
30. Cheng, J.; Huang, W.; Cao, S.; Yang, R.; Yang, W.; Yun, Z.; Wang, Z.; Feng, Q. Enhanced performance of brain tumor classification via tumor region augmentation and partition. *PLoS ONE* **2015**, *10*, e0140381. [[CrossRef](#)]
31. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. In Proceedings of the Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–8 December 2012; pp. 1–9.
32. He, K.; Zhang, Z.; Ren, S.; Sun, J. Deep residual learning for image recognition. *arXiv* **2015**, arXiv:1512.03385.
33. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, E.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. *arXiv* **2014**, arXiv:1409.4842v1.
34. Huang, G.; Liu, Z.; Van der Maaten, L.; Weinberger, K.Q. Densely Connected Convolutional Networks. *arXiv* **2018**, arXiv:1608.06993. [[CrossRef](#)]
35. Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; He, K. Aggregated residual transformations for deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 5987–5995. [[CrossRef](#)]
36. Noh, K.J.; Choi, J.; Hong, J.S.; Park, K.R. Finger-Vein Recognition Based on Densely Connected Convolutional Network Using Score-Level Fusion with Shape and Texture Images. *IEEE Access* **2020**, *8*, 96748–96766. [[CrossRef](#)]
37. Nguyen, K.; Fookes, C.; Ross, A.; Sridharan, S. Iris Recognition with off-the-Shelf CNN Features: A Deep Learning Perspective. *IEEE Access* **2018**, *6*, 18848–18855. [[CrossRef](#)]
38. Chang, C.-C.; Lin, C.-J. LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.* **2011**, *2*, 27. [[CrossRef](#)]
39. The Scikit-Learn Library for Machine Learning. Available online: <https://scikit-learn.org/stable/index.html> (accessed on 10 November 2022).
40. Sung, F.; Yang, Y.; Zhang, L.; Xiang, T.; Torr, P.H.S.; Hospedales, T.M. Learning to compare: Relation network for few-Shot learning. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 1199–1208. [[CrossRef](#)]
41. Yang, R.; Xu, X.; Li, X.; Wang, L.; Pu, F. Learning relation by graph neural network for SAR image few-shot learning. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, Waikoloa, HI, USA, 26 September–2 October 2020; pp. 1743–1746. [[CrossRef](#)]
42. Alajaji, D.; Alhichri, H.S.; Ammour, N.; Alajlan, N. Few-shot learning for remote sensing scene classification. In Proceedings of the 2020 Mediterranean and Middle-East Geoscience and Remote Sensing Symposium (M2GARSS), Tunis, Tunisia, 9–11 March 2020; pp. 81–84. [[CrossRef](#)]
43. Mateo, J.R.S.C. Weighted sum method and weighted product method. In *Multi Criteria Analysis in the Renewable Energy Industry; Green Energy and Technology*; Springer: London, UK, 2012.
44. Koo, J.H.; Cho, S.W.; Baek, N.R.; Park, K.R. Multimodal human recognition in significantly low illumination environment using modified EnlightenGAN. *Mathematics* **2021**, *9*, 1934. [[CrossRef](#)]
45. Tensorflow Library. Available online: <https://www.tensorflow.org/> (accessed on 1 October 2022).
46. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2017**, arXiv:1412.6980.

47. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient convolutional neural networks for mobile vision application. *arXiv* **2017**, arXiv:1704.04861.
48. Baid, U.; Ghodasara, S.; Mohan, S.; Bilello, M.; Calabrese, E.; Colak, E.; Farahani, K.; Kalpathy-Cramer, J.; Kitamura, F.C.; Pati, S.; et al. The RSNA-ASNR-MICCAI BraTS 2021 Benchmark on Brain Tumor Segmentation and Radiogenomic Classification. *arXiv* **2021**, arXiv:2107.02314.
49. Menze, B.H.; Jakab, A.; Bauer, S.; Kalpathy-Cramer, J.; Farahani, K.; Kirby, J.; Burren, Y.; Porz, N.; Slotboom, J.; Wiest, R.; et al. The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS). *IEEE Trans. Med. Imaging* **2015**, *34*, 1993–2024. [[CrossRef](#)]
50. Bakas, S.; Akbari, H.; Sotiras, A.; Bilello, M.; Rozycski, M.; Kirby, J.S.; Freymann, J.B.; Farahani, K.; Davatzikos, C. Advancing The Cancer Genome Atlas glioma MRI collections with expert segmentation labels and radiomic features. *Nat. Sci. Data* **2017**, *4*, 170117. [[CrossRef](#)]
51. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-CAM: Visual explanations from deep networks via gradient based localization. *arXiv* **2016**, arXiv:1610.02391v1.