

Article

# Multimodal Brain Tumor Classification Using Deep Learning and Robust Feature Selection: A Machine Learning Application for Radiologists

Muhammad Attique Khan <sup>1</sup>, Imran Ashraf <sup>2</sup>, Majed Alhaisoni <sup>3</sup>, Robertas Damaševičius <sup>4,5,\*</sup>, Rafal Scherer <sup>6</sup>, Amjad Rehman <sup>7</sup> and Syed Ahmad Chan Bukhari <sup>8</sup>

<sup>1</sup> Department of Computer Science, HITEC University, Museum Road, Taxila 47080, Pakistan; attique.khan440@gmail.com

<sup>2</sup> Department of Computer Engineering, HITEC University, Museum Road, Taxila 47080, Pakistan; imran.ashraf@hitecuni.edu.pk

<sup>3</sup> College of Computer Science and Engineering, University of Ha'il, Ha'il 81451, Saudi Arabia; m.alhaisoni@uoh.edu.sa

<sup>4</sup> Faculty of Applied Mathematics, Silesian University of Technology, 44-100 Gliwice, Poland

<sup>5</sup> Department of Applied Informatics, Vytautas Magnus University, 44404 Kaunas, Lithuania

<sup>6</sup> Department of Intelligent Computer Systems, Czestochowa University of Technology, 42-200 Czestochowa, Poland; rafal.scherer@pcz.pl

<sup>7</sup> College of Computer and Information Sciences, Prince Sultan University, Riyadh 11586, Saudi Arabia; rkamjad@gmail.com

<sup>8</sup> Division of Computer Science, Mathematics and Science, Collins College of Professional Studies, St. John's University, New York, NY 11439, USA; bukharis@stjohns.edu

\* Correspondence: robertas.damasevicius@polsl.pl

Received: 18 June 2020; Accepted: 4 August 2020; Published: 6 August 2020



**Abstract:** Manual identification of brain tumors is an error-prone and tedious process for radiologists; therefore, it is crucial to adopt an automated system. The binary classification process, such as malignant or benign is relatively trivial; whereas, the multimodal brain tumors classification (T1, T2, T1CE, and Flair) is a challenging task for radiologists. Here, we present an automated multimodal classification method using deep learning for brain tumor type classification. The proposed method consists of five core steps. In the first step, the linear contrast stretching is employed using edge-based histogram equalization and discrete cosine transform (DCT). In the second step, deep learning feature extraction is performed. By utilizing transfer learning, two pre-trained convolutional neural network (CNN) models, namely VGG16 and VGG19, were used for feature extraction. In the third step, a correntropy-based joint learning approach was implemented along with the extreme learning machine (ELM) for the selection of best features. In the fourth step, the partial least square (PLS)-based robust covariant features were fused in one matrix. The combined matrix was fed to ELM for final classification. The proposed method was validated on the BraTS datasets and an accuracy of 97.8%, 96.9%, 92.5% for BraTs2015, BraTs2017, and BraTs2018, respectively, was achieved.

**Keywords:** brain tumor; healthcare; linear contrast; transfer learning; deep learning features; feature selection; feature fusion; PLS; ELM

## 1. Introduction

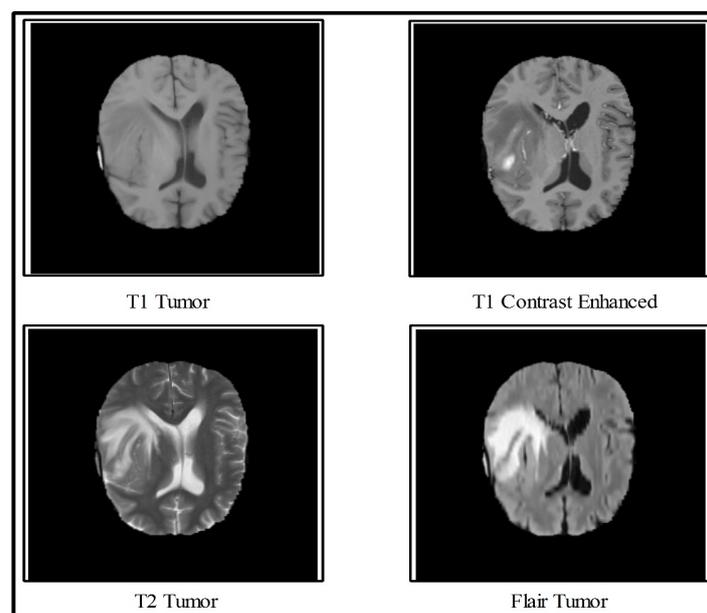
A brain tumor is an abnormal growth of brain cells in an uncontrollable way [1,2]. Brain tumors can be cancerous or noncancerous. The gravity inside the skull can accelerate the growth of a brain tumor. In the worst case, it can cause brain damage, which can be life-threatening. According to an

estimate, 18,020 adults will die from the primary cancerous brain and central nervous system (CNS) tumors in 2020 [3]. Various manifestations and classes of brain tumors have different appearances on magnetic resonance imaging (MRI) data [4,5]. Therefore, MRI scans are typically used to detect and classify brain tumors. MRI assists doctors in evaluating tumors in order to plan for further treatment. This treatment depends on various factors like shape, size, type, grade, and location of cancer. Depending on the patient's condition, these factors can have enormous variation. Hence, accurate recognition and classification of brain tumors are critical for proper treatment [6].

Manual identification of brain tumors and tracking their changes over time are tedious and error-prone activities [7]; hence, automated systems are required to replace the conventional manual methods. In the last decade, the results of deep neural networks (DNNs) exhibited excellent performance, which is also evident from the recent multimodal BraTS challenges [8]. Another well-known technique of deep learning is convolutional neural networks (CNN), which shows excellent performance both for 2D and 3D medical images [9,10]. Similarly, the transfer learning technique is typically utilized in case of limited availability of data and computational resources to save time [11]. This technique uses the knowledge acquired for one task to solve related ones [12]. Feature fusion is the detection of co-related features in order to fuse them to identify and compact a set of salient features to improve the detection accuracy. Furthermore, to reduce the time and space complexity, intelligent feature selection is typically required [13,14].

### 1.1. Significant Challenges and Motivation

In the multimodal brain tumor classification task, several challenges exist, which reduce automated system performance. In the classification task, two steps are usually performed—features extraction and classifiers-based classification. The features extraction is an essential step in pattern recognition, which predicts an object based on its crucial characteristics like shape, color, names, and a few more. However, the performance of classifiers depends on the strength of the extracted features. The recent success of deep learning in the medical domain reflects the interest of researchers of computer vision. However, all extracted deep learning features are normally not useful for correct classification and they might consume much time during the execution process. Moreover, the similarity among some tumors like T2 and Flair tumors is also very high, as demonstrated in Figure 1, which makes this classification process even more complicated. Further, the T1 tumors and T1 contrast-enhanced tumors have shallow contrast, which is another challenge for correct feature extraction.



**Figure 1.** Sample images collected from Multimodal Brain Dataset (BRATS) [15].

In this work, we propose a deep learning scheme for multimodal brain tumors classification. To handle the problem of shallow contrast, we implemented a linear contrast enhancement technique, which was further refined through histogram equalization. Transfer learning was used for feature extraction from two different CNN models and the fusion was performed. The motivation behind the fusion of two CNN models was to get a new feature vector with more information. Although this process improved accuracy, the computational time was affected. To further enhance efficiency and computational time, we proposed a feature selection technique. The robust features obtained using this technique were later classified through the Extreme Learning Machine (ELM).

### 1.2. Major Contributions

- We divided the image into two clusters based on a K-Means clustering algorithm and applied edge-based histogram equalization on each image. Further, the discrete cosine transform (DCT) was utilized for local information enhancement.
- Deep learning features were extracted from two pre-trained CNN models through transfer learning (TL). The last FC layer was used in both models for feature extraction.
- The Partial Least Square (PLS) based features of both CNN models were fused in one matrix.
- The robust features were selected using correntropy-based joint group learning. The robust features were finally classified using the ELM classifier.
- Three datasets such as BRATS 2015, BRATS 2017, and BRATS 2018 were used for the experiments and the statistical analysis to examine the scalability of the proposed classification scheme.

## 2. Related Work

Classification of multimodal brain tumors (i.e., T1, T2, T1CE, and Flair) required the determination of altered features, such as shape and texture in the MRI Image [16]. The popular method of diagnosis of these tumors—which spread widely among computer vision researchers—is a computer-aided diagnosis (CAD) system [1,17]. In a CAD system, two main stages are involved—first, tumor preprocessing and detection, and second, classifying the tumor into a relevant category. In this work, we focused on the classification task of multimodal brain tumors. For classification, we used the BRATS series based on few top submissions [18–21]. Amin et al. [22] introduced a CNN framework for brain tumor classification. In the presented method, the DWT fusion process was performed to improve the original MRI scan and then a partial diffusion filter was employed for noise removal. Later on, they used a global thresholding algorithm for tumor extraction that passed to the CNN model for classification of tumors into the related categories. Five BRATS datasets, namely, BRATS2012, 2013, 2015, 2018, and BRATS2013 were used and showed improved performance on the fusion approach. Sajjad et al. [23] presented a CNN-based multimodal tumor classification system. They initially segmented the tumor regions in the MRI scans using CNN. Then, they performed an extensive data augmentation to train a good CNN model. Later on, they fine-tuned the pre-trained CNN model using augmented brain data. The last layer was used as a classification of tumors in the presented method and it showed that augmented data gave better results on the selected datasets.

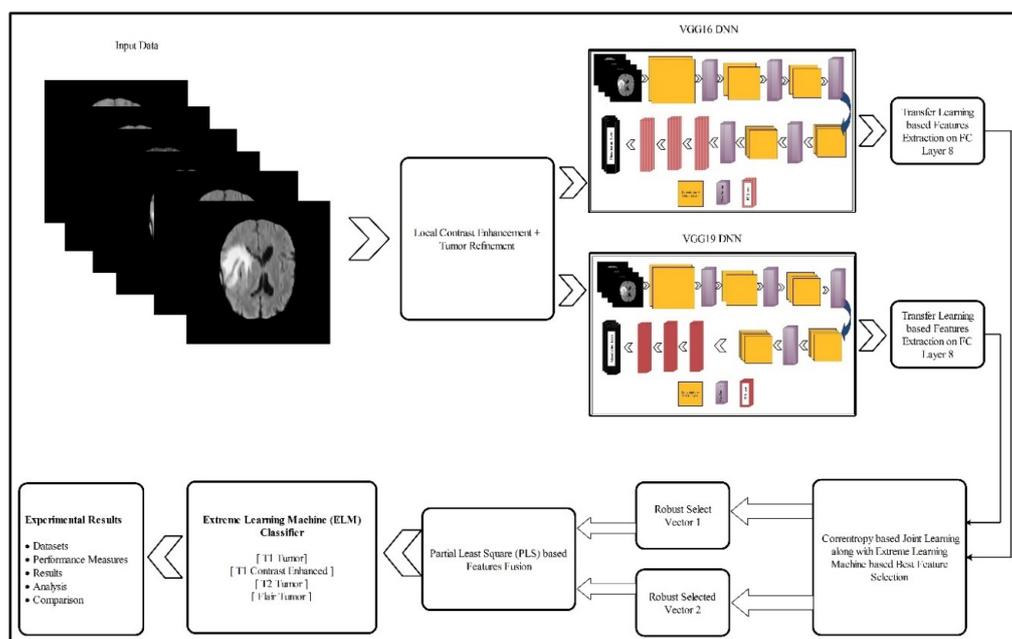
Sharif et al. [24] presented an active deep learning system for the segmentation and classification of brain tumors. They initially performed contrast enhancement, and the resultant image was passed to the Saliency-based Deep Learning (SbDL) method, for the construction of a saliency map. The thresholding was applied in the next step, and the resultant images were used to fine-tune the pre-trained CNN model Inception V3. Further, they also extracted dominant rotated local binary pattern (DRLBP) features, fused with CNN features. Later on, a PSO-based optimization was performed and the optimal vector was passed to the Softmax classifier for final classification. They used BRATS 2015, 2017, and 2018 datasets for evaluation, and achieved improved classification accuracy. In [25], the authors presented a CNN-based scheme for the classification of brain tumors. They considered the problem of structural variability of the tumor around the adjacent regions. For this purpose, they designed

small kernels to keep the weights of each neuron very small. Taking advantage of these weights, they achieved an accuracy of 97.5%.

Vijh et al. [26] presented an adaptive particle swarm optimization (PSO) with the Otsu method to find the optimal threshold value. Later, they applied anisotropic diffusion (AD) filtering on brain MRI images to cancel noise and improve image quality. Features were extracted from enhanced images that were used both for training the CNN and performing the classification. Other methods were also introduced in the literature for brain tumor classification, such as a generative adversarial network (GAN)-based approach [19], artificial neural network (ANN)-based learning [27], ELM-based learning [28], residual network [29], standard-features-based classification [30,31], adaptive independent subspace analysis [32], transfer learning-based tumors classification [33], and Excitation DNN [34]. In addition, Toğaçar et al. [35] proposed a hybrid method based on CNN and feature selection, for the classification of brain tumors. They achieved an improved accuracy of above 90%. In the above techniques, they did not provide the computational time. However, the computational time was most needed for this current era for each automated system. The more recent, Muhammad et al. [36] presented a detailed review on multi-grade brain tumor classification. They presented a detailed description of brain tumor classification (BTC) steps like preprocessing of tumor, deep learning features, and classification. They discussed detailed limitations and achievements of existing deep learning techniques for BTC. In addition, they also presented the importance of transfer learning for deep learning feature extraction.

### 3. Proposed Methodology

In this section, the proposed methodology for multimodal brain tumor classification using deep learning is presented. The proposed method consists of five core steps—linear contrast stretching, deep learning features extraction using transfer learning, a correntropy-based joint learning approach along with ELM for best features selection, the PLS-based fusion of the selected features, and finally the ELM-based classification. The testing of the proposed method was performed on the BRATS datasets. The performance of the approach was checked using standard performance measures like accuracy and false negative rate (FNR). Furthermore, the performance of the proposed work was also reported by measuring the execution time. A detailed flow of the proposed methodology is illustrated in Figure 2. In the following, the technical description of each step is provided.



**Figure 2.** Proposed architecture for multimodal brain tumor classification using deep learning features and extreme learning machine (ELM).

### 3.1. Linear Contrast Enhancement

Improving the graphic features of an image is the primary objective of contrast enhancement. It is a preprocessing step that is used in many applications like biomedical imaging, agriculture infections diagnosis, and some others [37–42]. The impact of low contrast images is not useful for feature extraction, as visually, tumors are not visible and error prone. Therefore, in this step, we improved the linear contrast of an image, which showed the main impact on the tumor region. For this purpose, we implemented a hybrid technique. In this technique, initially, we split the image into two parts using the K-Means clustering algorithm. Then, edge-based texture histogram equalization (HE) was applied. Later on, DCT was applied to combine both clusters in one image. The resulting image had enhanced contrast as compared to the original one. The mathematical formulation of this method is given as follows:

Consider, we have a dataset  $\Delta = \{\tau_1, \tau_2, \tau_3, \dots, \tau_N\}$ ,  $\tau_N \in \mathbb{R}^d$ . Consider  $\tau(x, y)$  is an MRI image of dimension  $N \times M$  where  $N = 256$  and  $M = 256$ , rows, and columns, respectively. Let  $\tau_i$  denotes the average of clusters  $K_i$  then using this, the criterion function is defined as follows:

$$\bar{S} = \sum_{i=1}^K \sum_{\tau \in K_i} |\tau - \tau_i|^2, \tag{1}$$

where  $\bar{S}$  denotes the sum of square error of all pixels,  $\tau_i$  means input images, and  $K$  implies the number of clusters that are initialized in this work. In K-Means, the Euclidean distance was used to as criterion distance, which was defined as follows:

$$D(\tau_i, y_i) = \sqrt{\sum_{i=1}^n (\tau_i - y_i)^2}, \tag{2}$$

where  $\tau_i$  and  $y_i$  are two vectors. This formulation obtained two clusters. Using resultant images defined by  $\tau_1(x, y)$ , we employed edge-based texture HE, where  $\tau_1(x, y) \in \bar{S}$ . For the resultant image  $\tau_1(x, y)$ , the gradient was computed as follows:

$$G(x, y) = \sqrt{G_x(x, y)^2 + G_y(x, y)^2}, \tag{3}$$

where  $G_x$  and  $G_y$  denotes  $x$  derivatives and  $y$  derivatives of  $\tau_1(x, y)$ , respectively. Later, the edge map was constructed using a threshold function, as follows:

$$EMp(x, y) = \begin{cases} 1 & G(x, y) < T \\ 0 & G(x, y) \geq T \end{cases} \tag{4}$$

From this equation, we considered the pixels with values higher than the threshold ( $T = 0.55$ ). These pixels were used for texture histogram computation (HC). Later on,  $\alpha$  and  $\beta$  were calculated, where  $\alpha$  denotes minimum and  $\beta$  denotes maximum pixel value. The grey levels whose value lied between  $\alpha$  and  $\beta$ , were represented as HC. Finally, the cumulative distribution function (CDF) and the transfer functions were applied to obtain an enhanced image. This was defined by Equations (5) and (6), as follows:

$$CDF(i) = \sum_{i=0}^n Th(i), \tag{5}$$

$$F_\tau = \tau_0 + (L - 1 - \tau_0)CDF, \tag{6}$$

The resultant image  $\tau_2(x, y) \in CDF(i) \& F_\tau$  was passed to the DCT method to refine the local contrast of the tumor region. Mathematically, this was computed as follows:

$$\tau_{xy} = \begin{cases} \frac{1}{\sqrt{M}} & x = 0, \quad 0 \leq y \leq M - 1 \\ \sqrt{\frac{2}{M}} \cos \frac{\pi(2y+1)x}{2M} & 1 \leq x \leq M - 1 \end{cases} \tag{7}$$

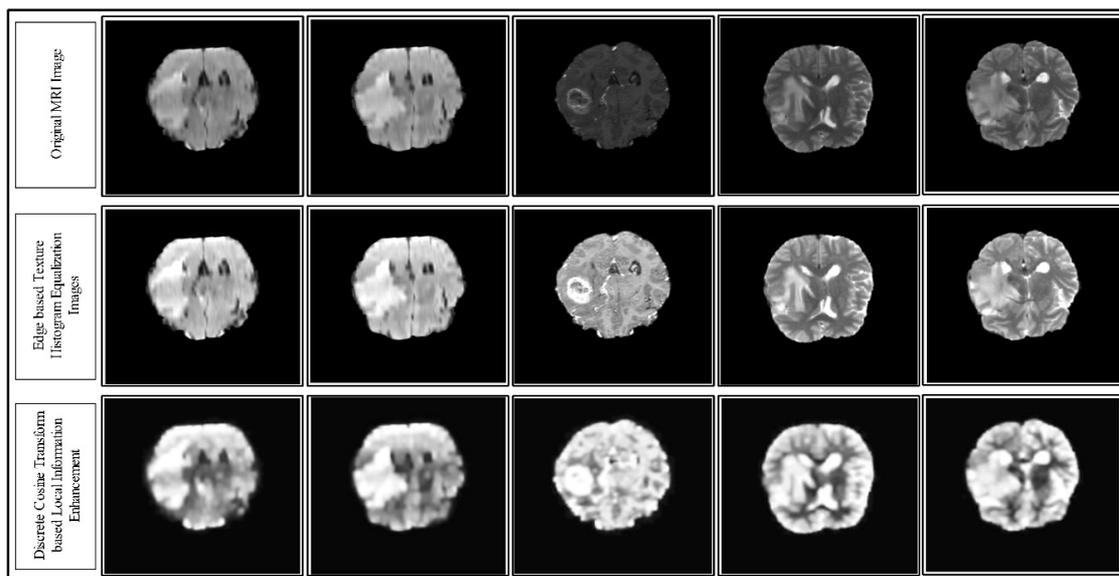
Hence, using  $\tau_{xy}$ , the DCT method was applied to an image  $\tau_2(x, y)$ , as follows:

$$\tau_3(x, y) = \tau_{xy} \times \tau_2 \times \tau_{xy}, \quad (8)$$

As  $\tau_{xy}$  is a real orthogonal matrix and its inverse could be computed as:

$$[\tau_3(x, y)]^{-1} = \tau_{xy} \times \tau_2 \times [\tau_{xy}]^t = 1, \quad (9)$$

where  $t$  denotes the transpose of an image. Hence, the representation of the final DCT enhanced image  $\tau_3(x, y)$  is depicted in Figure 3. In this figure, the sample enhancement results are presented for each step (top to bottom).



**Figure 3.** Results of the proposed hybrid local contrast enhancement using the MRI images of the Multimodal BRATS 2018 dataset. The results are shown from top to bottom.

### 3.2. Deep Learning Features

The deep learning features were extracted using two pre-trained deep CNN models—VGG16 and VGG19. The visual representation of both models is shown in Figures 4 and 5, respectively.

The VGG16 model consisted of 12 convolution layers, 15 ReLU activation layers, five max-pooling layers, three fully connected (FC) layers, and one Softmax layer, as a classification layer. The input layer size was  $224 \times 224 \times 3$ . The number of filters in the first convolution layer was 64, and the filter size was  $3 \times 3 \times 3$ , along with a stride of  $1 \times 1$ . In the next convolution layer, the number of filters was not updated but the filter size was updated to  $3 \times 3 \times 64$ . Further, the dimension of learnable weights was  $3 \times 3 \times 64 \times 64$ , which were  $3 \times 3 \times 3 \times 64$  in the first convolution layer. The learnable weights of each convolution layer were updated according to the number of filters and the filter size. In the first max-pooling layer, a  $2 \times 2$  filter size was opted along with the same stride  $2 \times 2$ . After the convolution layers, three FC layers were added. The learnable weights dimension of the first FC layer was  $4096 \times 25088$ . After a 50% dropout, the weights matrix size of the second FC layer was  $4096 \times 4096$ . Another dropout layer was added and a ratio of 50% was set. The resultant weight matrix used as an input of the third layer (denoted as FC8) returned a weight matrix of dimension  $1000 \times 4096$ . Finally, the Softmax function and the classification layers were added for the final classification.

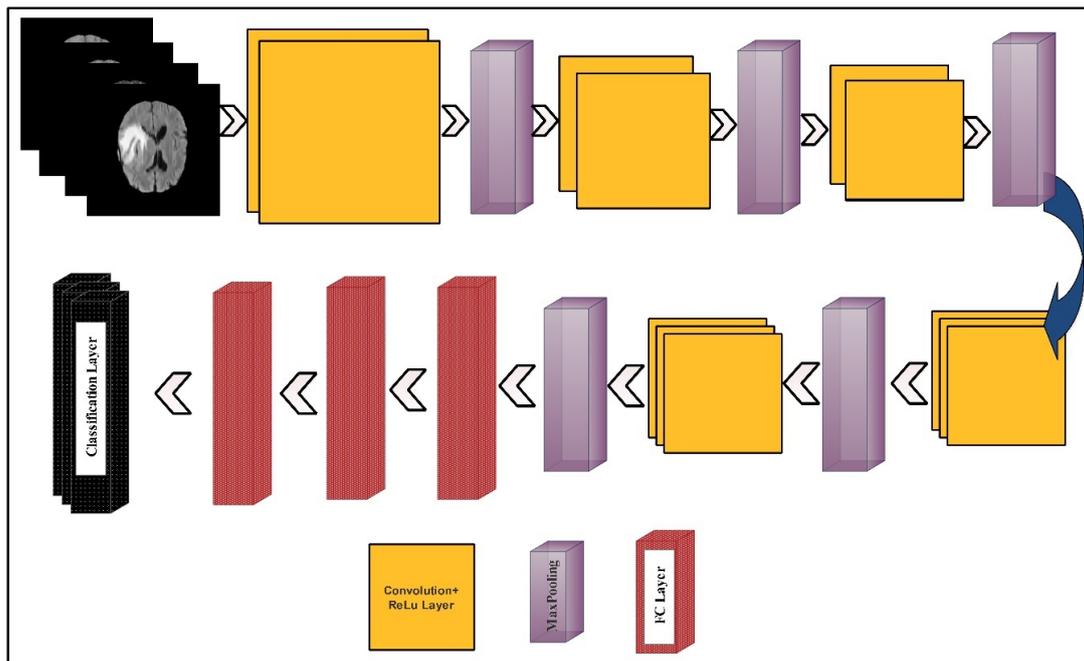


Figure 4. A layered wise architecture of the VGG16 deep learning model.

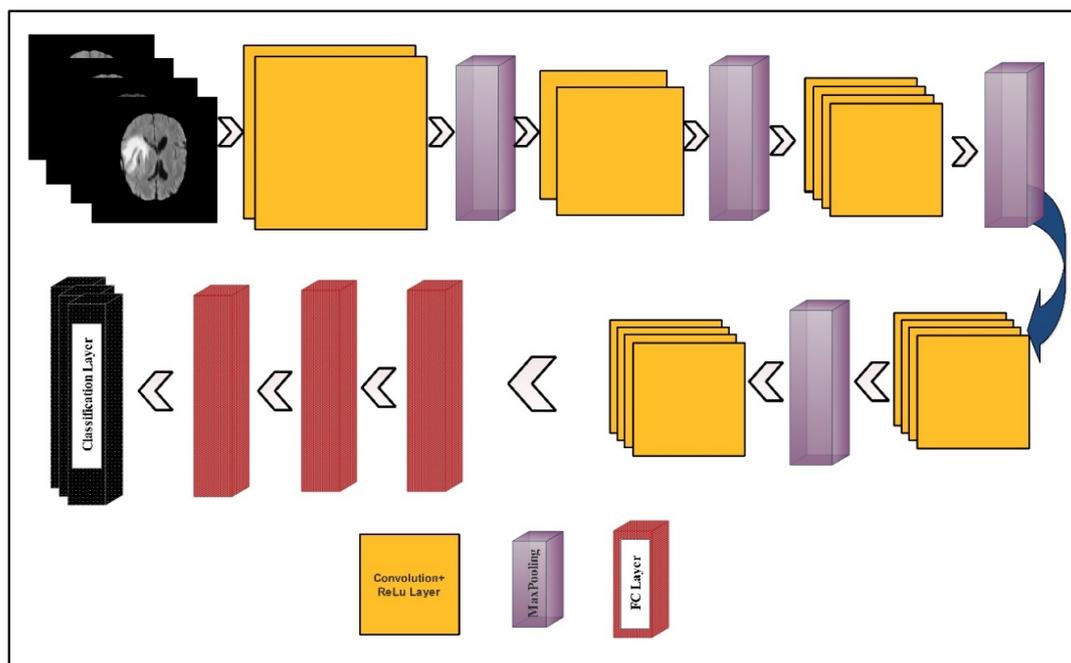


Figure 5. A layered wise architecture of the VGG19 deep learning model.

VGG19 model consists of a series of 16 convolution layers, 19 ReLu activation layers, four max-pooling layers, three FC layers, and one Softmax layer as a classification layer. The input layer size was  $224 \times 224 \times 3$ . The number of filters in the first convolution layer was 64, and the filter size was  $3 \times 3 \times 3$ . This filter size was updated according to the number of filters. In the first max-pooling layer, a  $2 \times 2$  filter size was opted along with the same stride. After the convolution layers, three FC layers were added. The weights dimension of the first FC layer was  $4096 \times 25088$ . After a 50% dropout, the weights matrix size of the second FC layer was  $4096 \times 4096$ . The resultant weight matrix used as an input of the third layer (denoted as FC8) returned a weight matrix of dimension  $1000 \times 4096$ .

### 3.3. Network Modification for Transfer Learning

Using domain adaptation transfer learning (TL) [43], we retrained both models (VGG16 and VGG19) on the BRATS datasets, without changes in any parameters. In the tuning process, first, we loaded the brain datasets and set the training/testing ratio to 60:40. Further, the labels of each image were also defined. Then, we set input and output layers for training. This process was conducted for both deep learning models. In this paper, for the VGG16 model, the input convolution layer (conv\_1) was employed, where the number of filters was 64, and the filter size was  $3 \times 3 \times 64$ . The selected output layer was FC8. Then, we performed activation on this layer and trained a new modified CNN network that included only the brain image features. The last two layers, namely, the classification and Softmax layers were removed. In the output, the resultant learnable weights vector length was  $4 \times 4096$ , and the feature-length was  $1 \times 1000$ . Hence, for  $n$  images, the feature vector length was  $N \times 1000$ , denoted by  $\eta_i$ . Similarly, for the VGG19 model, the last two layers were removed. The convolution layer (conv\_1) was employed as an input with 64 filters, and the filter size was  $3 \times 3 \times 64$ . The selected output layer was FC8, which we chose for the activation function. The activation function was performed on this layer and trained a new modified CNN network that included only the brain image features. The dimension of the learnable weight matrix was  $4 \times 4096$ , and the length of the extracted feature vector was  $1 \times 1000$ . For  $n$  brain images, the feature vector length should be  $N \times 1000$  and should be denoted by  $\eta_j$ .

### 3.4. Feature Selection

The main motive of the feature selection step was to remove the redundancy among features and select only those features that were robust for the correct classification. The second motive of this step was to minimize the number of predictors, which helped in the fast execution of the testing process. To inspire with these two essential functionalities, we implemented a technique named correntropy via mutual learning and ELM (CML-ELM). The working of this method is presented in Algorithm 1:

---

**Algorithm 1** Proposed feature selection method using CML-ELM.

---

**Input:**  $\eta_i, \eta_i \in \{\eta_1, \eta_2, \eta_3, \dots, \eta_i\}$

**Output:**  $S_w(i), S_w(i) \in \{S_w(1), S_w(2), \dots, S_w(i)\}$

**Start**

**Step 1: Parameters Initialization**

$$\begin{aligned} S_w(1) &= \eta_i, i = 0, 1, 2, 3, \dots, n \\ \alpha_{-1} &= 0 \\ \alpha_0 &= 1 \\ LR &= LR_0 \end{aligned}$$

**Step 2: For  $i = 1$  to  $K$**

**do**

$$\begin{aligned} b_i &= \frac{\alpha_{i-2}-1}{\alpha_{i-1}} \\ A^i &= S_w(i) + b_i(S_w(i) - S_w(i-1)) \end{aligned}$$

**Step 3: Update  $S_w(i+1)$**

**Step 4: Find the minimum value of  $LR$  among  $(LR_{(i-1)},$**

**$2LR_{(i-1)}, 3LR_{(i-1)}, \dots$ )**

**$:- f(S_w(i)) \leq g_{(LR_i, U^i)} S_w(i+1)$**

**Step 5: Passed computed  $LR$  values in ELM classifier**

**Step 6: Find MSER for ELM classifier**

**Step 7: If  $MSER \geq 0.1$**

**Update  $LR_{i+1}$**

**Step 8:**  $\alpha_{i+1} = \frac{1 + \sqrt{1 + 4^2}}{2}$

**End For**

**End**

---

In the above algorithm, the notation  $\eta_i$  denotes the original feature vector of the VGG16 deep learning model,  $S_w(i)$  means selected feature vector,  $LR$  denotes regularization parameter,  $b_i$  is a selected parameter,  $A^i$  is an affine combination of  $S_w(i)$  and  $S_w(i - 1)$ , MSER denotes mean squared error, computed by Equation (10), and the updating of features  $S_w(i + 1)$  are done by Equation (11).

$$MSEER = \frac{1}{n} \sum_{i=1}^K (LR_i - \widehat{LR}_i)^2, \tag{10}$$

$$S_w(i + 1) = \operatorname{argmin}_{S_w} \frac{1}{2} \|S_w - V\|_F^2 - \frac{1}{LR_i} f_{ns}(S_w), \tag{11}$$

$$S_w(i + 1) = \operatorname{argmin}_{S_w(1), \dots, S_w(D)} \frac{1}{2} \sum_{j=1}^D \|S_w(j) - V_j\|_2^2 + \frac{p_0}{LR_i} f_{ns} \|S_w(j)\|_2, \tag{12}$$

$$V = A^i - \frac{1}{LR_i f'(A^i)}, \tag{13}$$

where the  $LR_i$  denotes the observed features, and  $\widehat{LR}_i$  denotes the predicted features. Each time, the MSER was calculated, and if its value was greater than or equal to 0.1, then the features were updated, iterating this process for 1000 times. If the target was not achieved, then the last iteration features were selected for the classification. Finally, a robust vector was obtained, where the dimension of this vector was  $X_1 \times K$  and was denoted by  $\eta_{S_w}(1)$ , where the  $K$  stood for the number of selected features and  $X_1$  denoted the total number of images. This feature selection process was also performed for the VGG19 feature vector  $\eta_j$  and obtained a robust feature vector of dimension  $X_2 \times K$  and denoted by  $\eta_{S_w}(2)$ , where  $X_2$  was the number of observations, and  $K$  represented the number of selected features.

### 3.5. Feature Fusion and Classification

Finally, the selected feature vectors were fused in one matrix using the PLS-based fusion approach. Consider  $\eta_{S_w}(1)$  and  $\eta_{S_w}(2)$  are two selected feature vectors of dimension  $X_1 \times K$  and  $X_2 \times K$ . Suppose  $\eta_{S_w}(j)$  represents a fused vector of dimension  $X_3 \times K$ . Further, we assumed that the central variables  $\vec{U}$  and  $\vec{V}$  were zero mean, where  $\vec{U} \in \eta_{S_w}(1)$  and  $\vec{V} \in \eta_{S_w}(2)$ . Let  $\delta_{uv} = \vec{U}\vec{V}$  and  $\delta_{vu} = \delta_{uv}^T \left( \left( \frac{1}{n} - 1 \right) \delta_{uv} \right)$  represent between set covariance among vectors  $\vec{U}$  and  $\vec{V}$ . The PLS held correlated features for fusion. Further, the fusion process through PLS also minimized the number of predictors. Mathematically, the decomposition method among  $\vec{U}$  and  $\vec{V}$  was defined as follows:

$$\vec{U} = \sum_{i=1}^d \eta_i \eta_{S_w}(1i)^T = E, \tag{14}$$

$$\vec{V} = \sum_{i=1}^d \eta_i \eta_{S_w}(2i)^T = F, \tag{15}$$

When using PLS, a pair of directions among  $u_i$  and  $v_i$  was found, as follows:

$$\{u_i; v_i\} = \operatorname{arg} \max_{u^T u = v^T v = 1} \operatorname{Cov} \left( \vec{U}^T u, \vec{V}^T v \right), \tag{16}$$

$$\{u_i; v_i\} = \operatorname{argmax}_{u^T u = v^T v = 1} u^T \delta_{uv} v, \text{ for } i = 1, 2, 3, \dots, d \text{ a } = 1, \tag{17}$$

These pairs were combined in one matrix and a resultant vector was obtained with  $X_3 \times K$  dimension. The fused vector was represented by  $\eta_{S_w}(j)$ . Later on, this vector was passed to ELM [44] for the final classification. The formulation of ELM was given as follows. For  $L$  hidden layers node, the activation function  $g(x)$  was defined as follows:

$$\sum_{i=1}^L \beta_i g_i(u_i) = \sum_{j=1}^L \beta_j g(u_i \cdot u_j + B_j), \tag{18}$$

$$\beta^T = O, \tag{19}$$

where  $L$  denotes a hidden layer, which was initialized as one in this work,  $\beta_i$  denotes the output weight vector,  $u_i$  is the input weight vector coming to the hidden layer,  $B_i$  denotes the offset value,  $H$  is the output hidden layer node,  $u_i \cdot u_j$  means an inner product of  $u_i$ , and  $O$  is the expected output. Equation (19) was solved as:

$$\hat{\beta}_{ELM} = \underset{\beta}{\operatorname{argmin}} \|\beta^T H - O\|, \tag{20}$$

To further improve the stability of ELM, we defined a minimization function as:

$$\min_w \frac{1}{2} \|\beta\|^2 + \frac{1}{2} c \sum_{i=1}^N \|\epsilon_i\|^2 \text{ s.t. } \beta^T h(u_i) = t_i - \epsilon_i, \tag{21}$$

where  $\epsilon_i$  denotes training error,  $t_i$  indicates corresponding labels to the sample  $u_i$ , and  $c$  denotes the penalty parameter. The labeled results of the proposed architecture are given in Figure 6.

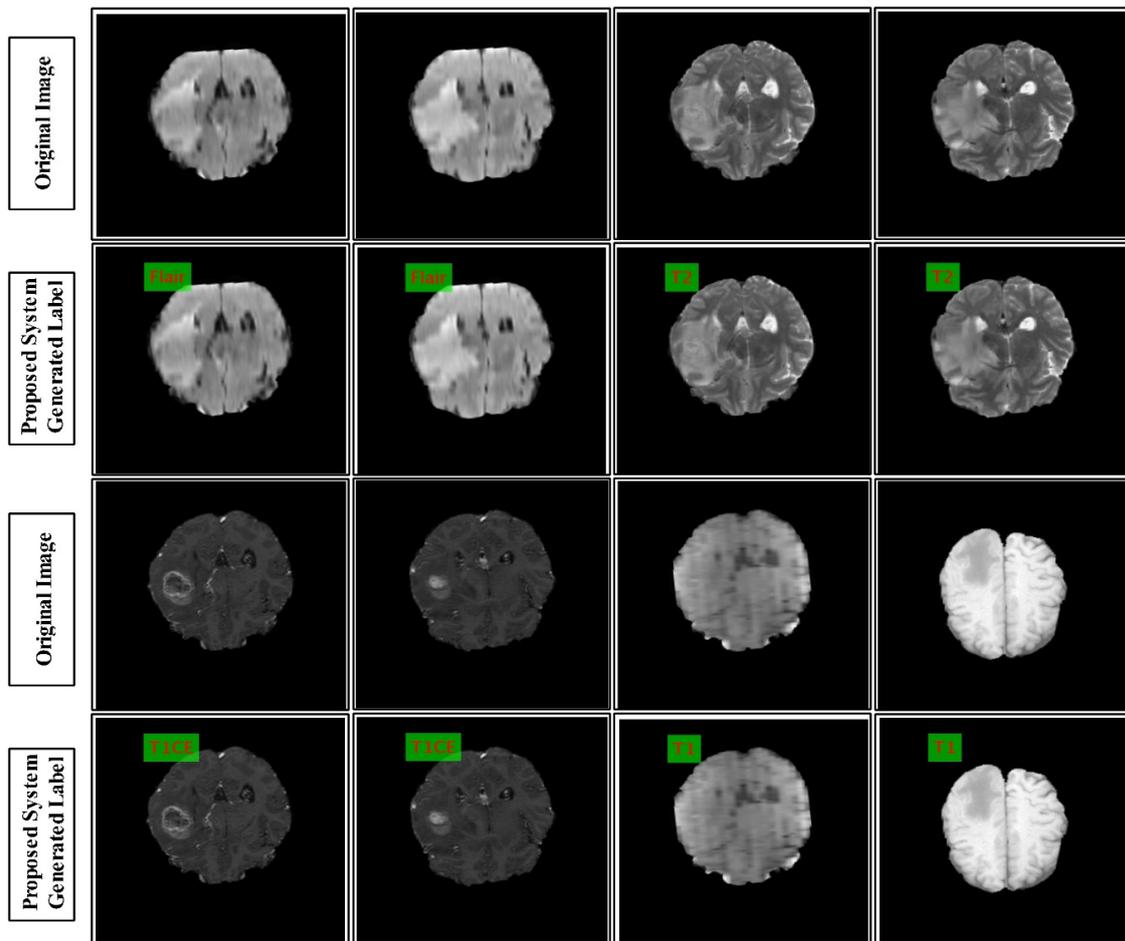


Figure 6. Labeled results generated by the proposed system. The label is shown in red on green square.

### 3.6. Experimental Results and Analysis

We present the classification results for the proposed ELM classifier using three datasets, namely, BraTS 2015, BraTS 2017, and BraTS 2018. For all datasets, a 60–40 split ratio was used along with 10-fold cross-validation. The results are provided for two different pipeline procedures, namely;

(i) feature extraction from FC layer seven and a performed feature selection approach that followed the feature fusion and classification and (ii) which followed the proposed architecture, as given in Figure 2. For the sake of comparison, we also provided the results for four well-known classifiers, like Naïve Bayes, Multiclass Support Vector Machine (MSVM), Softmax, and Ensemble Tree, as baselines. The performance of all classifiers was validated by the following measures, namely accuracy and FNR measures. Furthermore, the clock time taken by each classifier was also reported to give the reader an idea about the classification time during the testing process. All simulations of the proposed technique were conducted on MATLAB 2019b (MathWorks, Natick, MA, USA). The personal Desktop Computer with 16 GB RAM and 128 GB SSD was used for these experiments. A graphics processing unit (GPU) was also utilized for feature extraction and classification, which significantly helped in improving the classification time. The execution time was also noted during the testing process; however, it was not consistent and was only based on the execution platform.

### 3.7. Results for the BraTS 2015 Dataset

Table 1 presents the classification results for the BraTS 2015 dataset. The results were provided for the proposed classifier, as well as the existing well-known classifiers, such as Naïve Bayes, MSVM, Softmax, and Ensemble Tree. These results were provided for two experimental pipeline procedures, as mentioned above. Apart from the validation measures in terms of accuracy and FNR, the results were also provided for the classification time in seconds. The entries in the bold represent the best results. It can be seen from Table 1 that the minimum accuracy achieved was 91.48% for Softmax. The maximum accuracy of 98.16% (FNR = 1.74%) was achieved by the ELM classifier, which used the proposed method.

**Table 1.** Classification results for the BraTS 2015 dataset.

Classifier	Feature Selection Technique	Validation Measures		
		Accuracy (%)	FNR (%)	Testing Time (s)
Naïve Bayes	Pro-FC7	93.29	6.71	117.68
	Proposed	94.19	5.81	104.02
MSVM	Pro-FC7	92.59	7.41	136.31
	Proposed	94.66	5.34	101.66
Softmax	Pro-FC7	91.48	8.52	96.69
	Proposed	93.98	6.02	<b>81.02</b>
Ensemble Tree	Pro-FC7	92.43	7.57	137.60
	Proposed	95.67	4.33	104.59
ELM	Pro-FC7	96.02	3.98	99.42
	Proposed	<b>98.16</b>	<b>1.74</b>	87.41

Pro-FC7 defines feature extraction from the FC7 layer and performed feature selection, as well as fusion, and 'Proposed' denotes the proposed classifier architecture, as given in Figure 2. The best values are shown in bold.

The proposed selection scheme also reduced the classification time during the testing process. In Table 1, time is given for all classifiers, which clearly shows that the time for the proposed method was lesser than that compared to Pro-FC7. The classification time for Softmax was minimum (81.02 s), using the proposed method. Though the classification time for the proposed classifier was not minimum (87.41 s), it was still quite close to Softmax and considerably lower, as compared to the rest of classifiers.

The results of the proposed method on the ELM classifier were also verified by the confusion matrix values presented in Table 2. The diagonal values showed the correct classification rate of each tumor class. The maximum achieved accuracy of Pro-FC7 was 96.02% for ELM (Table 1), which could also be verified by the confusion matrix in Table 3.

**Table 2.** Confusion matrix of the proposed method for the BraTS 2017 dataset (Proposed).

Class	T1	T1CE	T2	Flair
T1	98.42%	<1%	0%	<1%
T1CE	1%	96.00%	2%	1%
T2	0%	0%	99.46%	<1%
Flair	<1%	<1%	<1%	98.80%

Grey background shows accuracy rate.

**Table 3.** Confusion matrix of ELM using Pro-FC7 approach for the BraTS 2017 dataset.

Class	T1	T1CE	T2	Flair
T1	97.16%	<1%	2%	0%
T1CE	<1%	95.24%	3%	1%
T2	<1%	2%	97.60%	0%
Flair	0%	2%	3%	94.00%

Grey background shows accuracy rate.

### 3.8. Results for the BraTS 2017 Dataset

Table 4 presents the classification results for the BraTS 2017 dataset. Results are provided for the proposed method along with several other well-known classifiers, such as Naïve Bayes, MSVM, Softmax, and Ensemble Tree. These results were provided for two experimental pipeline procedures, as mentioned above. Apart from the validation measures in terms of accuracy and FNR, results were also provided for the classification time in seconds. It can be clearly seen from Table 4 that the ELM classifier, which used the proposed method, had an accuracy of 97.26% and an FNR of 2.74%. The minimum met accuracy was 90.09% for Softmax.

**Table 4.** Classification results for the BraTS 2017 dataset.

Classifier	Feature Selection Technique	Validation Measure		
		Accuracy (%)	FNR (%)	Testing Time (s)
Naïve Bayes	Pro-FC7	91.59	8.41	197.46
	Proposed	93.66	6.34	104.59
MSVM	Pro-FC7	90.09	9.91	211.62
	Proposed	94.58	5.42	171.42
Softmax	Pro-FC7	91.67	8.33	111.44
	Proposed	93.98	6.02	91.25
Ensemble Tree	Pro-FC7	93.69	6.31	147.38
	Proposed	95.42	4.58	101.29
ELM	Pro-FC7	95.82	4.18	107.59
	<b>Proposed</b>	<b>97.26</b>	<b>2.74</b>	<b>89.64</b>

The best values are shown in bold.

The proposed selection scheme also reduced the classification time during the testing process, as was evident from the results shown in the last column in Table 4. The classification time for ELM was minimum (89.64 sec) using the proposed method, which clearly showed the improved efficiency of the ELM classifier.

The results of the proposed method on the ELM classifier could also be verified by the confusion matrix in Table 5. The diagonal values showed the correct classification rate of each tumor class, which were 96.24%, 98.66%, 97.20%, and 97% for the T1, T1CE, T2, and Flair tumors. The maximum achieved accuracy of Pro-FC7 was 95.82% for ELM, which could be further verified by the results in Table 6.

**Table 5.** Confusion matrix of the proposed method for the BraTS 2017 dataset.

Class	T1	T1CE	T2	Flair
T1	96.24%	2%	1%	<1%
T1CE	<1%	98.66%	0%	1%
T2	2%	0%	97.20	<1%
Flair	1%	0%	2%	97.00%

Grey background shows accuracy rate.

**Table 6.** Confusion matrix of ELM using the Pro-FC7 approach for the BraTS 2017 dataset.

Class	T1	T1 CE	T2	Flair
T1	94.20%	4%	1%	<1%
T1 CE	4%	94.84%	3%	2%
T2	0%	3%	96.68%	<1%
Flair	<1%	1%	2%	96.02%

Grey background shows accuracy rate.

### 3.9. Results of the BraTS 2018 Dataset

Table 7 presents the classification results for the BraTS 2018 dataset. Results were provided for the proposed method, as well as for other well-known classifiers, such as Naïve Bayes, MSVM, Softmax, and Ensemble Tree. These results were provided for two experimental pipeline procedures, as discussed earlier in Section 3. It can be seen from this table that the maximum achieved accuracy was 93.40% for the ELM classifier, using the proposed method. The noted FNR rate was 6.60%. The minimum achieved accuracy was 89.49%, using the proposed method for the Naïve Bayes classifier.

**Table 7.** Classification results for the BraTS 2018 Dataset.

Classifier	Feature Selection Technique	Validation Measure		
		Accuracy	FNR	Testing Time (s)
Naïve Bayes	Pro-FC7	87.63	12.37	204.31
	Proposed	89.49	10.51	117.62
MSVM	Pro-FC7	88.19	11.81	207.56
	Proposed	91.34	8.66	167.49
Softmax	Pro-FC7	90.26	9.74	131.31
	Proposed	92.42	7.58	91.63
Ensemble Tree	Pro-FC7	89.16	10.84	151.34
	Proposed	91.79	8.21	106.12
ELM	Pro-FC7	91.69	8.31	97.04
	<b>Proposed</b>	<b>93.40</b>	<b>6.60</b>	<b>63.83</b>

The best values are shown in bold.

The classification accuracy was also computed for Pro-FC7 to analyze the proposed results. For Pro-FC7, the maximum achieved accuracy was 91.69% for the ELM classifier. The accuracy of ELM using the proposed method and Pro-FC7 was further verified through Tables 8 and 9. In both these tables, the diagonal values represent the correct predicted rate of each tumor class, such as T1, T2, T1CE, and Flair.

Time was measured for each classifier during the testing process and presented in Table 7. We used tic-toc commands to compute the testing computational time of proposed method. In this table, it was observed that the best execution time was (63.83 s) for the ELM classifier, using the proposed method. However, this time was based on the platform that was used like GPU, system RAM, etc. Based on the presented results of accuracy and the testing execution time, the effectiveness of the proposed method was apparent for the accurate and efficient brain tumor type classification.

**Table 8.** Confusion matrix of the proposed method for the BraTS 2017 dataset.

Class	T1	T1CE	T2	Flair
T1	89.40%	7%	<1%	3%
T1CE	3%	94.60%	<1%	2%
T2	2%	4%	93.20%	<1%
Flair	1%	0%	3%	96.00%

Grey background shows accuracy rate.

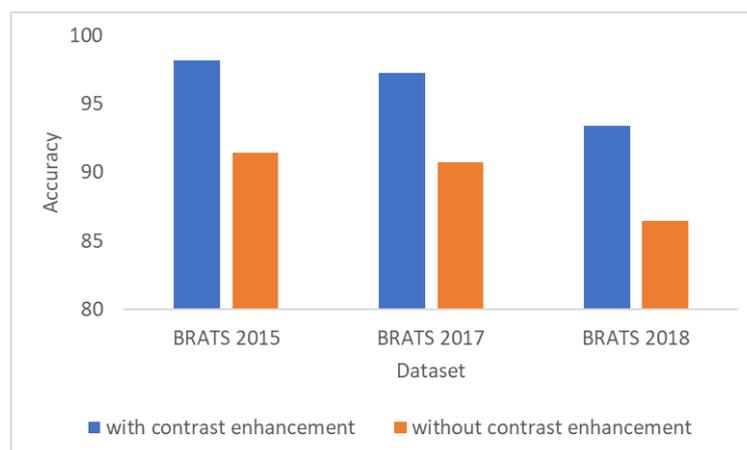
**Table 9.** Confusion matrix of ELM using the Pro-FC7 approach for the BraTS 2017 dataset.

Class	T1	T1CE	T2	Flair
T1	88.18%	6%	3%	<3%
T1CE	<1%	92.16%	6%	1%
T2	1%	7%	91.62%	<1%
Flair	2%	<1%	5%	92.14%

Grey background shows accuracy rate.

### 3.10. Results of the Contrast Enhancement

Figure 7 shows the importance of the contrast enhancement step. In this figure, it can be seen that, if the contrast enhancement step is not employed, the results show a decrease of almost 7% of accuracy for all BraTS datasets.

**Figure 7.** Accuracy results with and without employing the contrast enhancement step.

### 3.11. Statistical Analysis of Results

To examine the stability of the proposed method results, a detailed statistical analysis was conducted in terms of variance, standard deviation, and standard error mean (SEM). The noted values were obtained after 1000 iterations. The detailed analysis of the proposed method for the BraTs2015 dataset is given in Table 10. In this table, the accuracy of ELM had low variability, and SEM was 0.1862, which was better than that compared to other methods. Table 11 shows the detailed analysis of the proposed method using the BraTs2017 dataset. The accuracy of ELM was better than that compared to other listed classifiers (SEM is 0.0754). Table 12 illustrates the analysis results for the BraTs2018 dataset. Here, the SEM for the proposed method was 0.2875. As compared to other classifiers, it was better, and the results were stable after the selected iterations. Overall, the results of the proposed method were more stable for all listed classifiers. Moreover, we also plotted the confidence interval of ELM at different confidence levels (CL), such as 90%, 95%, 99%, etc., as shown in Figures 8–10. As shown in Figure 8, at 95% CL, the margin of error was  $97.763 \pm 0.365$  ( $\pm 0.37\%$ ). Similarly, in Figures 9 and 10, the margin of error at 95% CL was  $97.1 \pm 0.148$  ( $\pm 0.15\%$ ) and  $92.79 \pm 0.564$  ( $\pm 0.61\%$ ), respectively.

Based on these values, it was shown that our method was significantly better than that compared to other classifiers.

**Table 10.** Detailed statistical analysis of the proposed method using the BraTS 2015 dataset.

Method	Min (%)	Avg (%)	Max (%)	$\sigma^2$	$\sigma$	SEM
Naïve Bayes	92.47	93.33	94.9	0.493	0.7021	0.4054
MSVM	92.19	93.42	96.66	1.016	1.0083	0.5821
Softmax	91.63	92.8	93.98	0.920	0.9593	0.5539
ET	92.98	94.32	95.67	1.206	1.0981	0.6340
<b>ELM</b>	<b>97.37</b>	<b>97.76</b>	<b>98.16</b>	<b>0.104</b>	<b>0.3225</b>	<b>0.1862</b>

The best values are shown in bold. Min, Avg, and Max are the minimum, average, and maximum accuracy, respectively. SEM—standard error of mean.

**Table 11.** Detailed statistical analysis of the proposed method using the BraTS 2017 dataset.

Method	Min (%)	Avg (%)	Max (%)	$\sigma^2$	$\sigma$	SEM
Naïve Bayes	91.04	92.35	93.66	1.144	1.0696	0.6175
MSVM	92.67	93.62	94.58	0.608	0.7797	0.451
Softmax	90.29	92.13	93.98	2.2693	1.5064	0.8697
ET	93.16	94.29	95.42	0.8512	0.9226	0.5326
<b>ELM</b>	<b>96.94</b>	<b>97.1</b>	<b>97.26</b>	<b>0.017</b>	<b>0.1306</b>	<b>0.0754</b>

The best values are shown in bold. Min, Avg, and Max are the minimum, average, and maximum accuracy, respectively. SEM—standard error mean.

**Table 12.** Detailed statistical analysis of the proposed method using the BraTS 2018 dataset.

Method	Min (%)	Avg (%)	Max (%)	$\sigma^2$	$\sigma$	SEM
Naïve Bayes	87.42	88.45	89.49	0.7141	0.8450	0.4879
MSVM	86.69	90.01	91.34	1.1704	1.0818	0.6246
Softmax	91.04	91.73	92.42	0.3174	0.5633	0.3252
ET	88.64	90.21	91.79	1.6537	1.2859	0.7424
<b>ELM</b>	<b>92.18</b>	<b>92.79</b>	<b>93.40</b>	<b>0.2480</b>	<b>0.4980</b>	<b>0.2875</b>

The best values are shown in bold. Min, Avg, and Max are the minimum, average, and maximum accuracy, respectively. SEM—standard error mean.

Confidence Level	Margin of Error	Error Bar
68.3%, $\sigma_{\bar{x}}$	97.7633 ±0.186 (±0.19%)	
90%, 1.645 $\sigma_{\bar{x}}$	97.7633 ±0.306 (±0.31%)	
95%, 1.960 $\sigma_{\bar{x}}$	97.7633 ±0.365 (±0.37%)	
99%, 2.576 $\sigma_{\bar{x}}$	97.7633 ±0.48 (±0.49%)	
99.9%, 3.291 $\sigma_{\bar{x}}$	97.7633 ±0.613 (±0.63%)	
99.99%, 3.891 $\sigma_{\bar{x}}$	97.7633 ±0.725 (±0.74%)	
99.999%, 4.417 $\sigma_{\bar{x}}$	97.7633 ±0.822 (±0.84%)	
99.9999%, 4.892 $\sigma_{\bar{x}}$	97.7633 ±0.911 (±0.93%)	

**Figure 8.** Confidence interval of statistical test for different critical values of *t*-test (BraTS 2015 dataset).

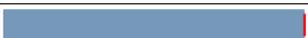
Confidence Level	Margin of Error	Error Bar
68.3%, $\sigma_{\bar{x}}$	97.1 $\pm$ 0.0754 ( $\pm$ 0.08%)	
90%, 1.645 $\sigma_{\bar{x}}$	97.1 $\pm$ 0.124 ( $\pm$ 0.13%)	
95%, 1.960 $\sigma_{\bar{x}}$	97.1 $\pm$ 0.148 ( $\pm$ 0.15%)	
99%, 2.576 $\sigma_{\bar{x}}$	97.1 $\pm$ 0.194 ( $\pm$ 0.20%)	
99.9%, 3.291 $\sigma_{\bar{x}}$	97.1 $\pm$ 0.248 ( $\pm$ 0.26%)	
99.99%, 3.891 $\sigma_{\bar{x}}$	97.1 $\pm$ 0.293 ( $\pm$ 0.30%)	
99.999%, 4.417 $\sigma_{\bar{x}}$	97.1 $\pm$ 0.333 ( $\pm$ 0.34%)	
99.9999%, 4.892 $\sigma_{\bar{x}}$	97.1 $\pm$ 0.369 ( $\pm$ 0.38%)	

Figure 9. Confidence interval of statistical test for different critical value of *t*-test (BraTS 2017 dataset).

Confidence Level	Margin of Error	Error Bar
68.3%, $\sigma_{\bar{x}}$	92.79 $\pm$ 0.288 ( $\pm$ 0.31%)	
90%, 1.645 $\sigma_{\bar{x}}$	92.79 $\pm$ 0.473 ( $\pm$ 0.51%)	
95%, 1.960 $\sigma_{\bar{x}}$	92.79 $\pm$ 0.564 ( $\pm$ 0.61%)	
99%, 2.576 $\sigma_{\bar{x}}$	92.79 $\pm$ 0.741 ( $\pm$ 0.80%)	
99.9%, 3.291 $\sigma_{\bar{x}}$	92.79 $\pm$ 0.946 ( $\pm$ 1.02%)	
99.99%, 3.891 $\sigma_{\bar{x}}$	92.79 $\pm$ 1.119 ( $\pm$ 1.21%)	
99.999%, 4.417 $\sigma_{\bar{x}}$	92.79 $\pm$ 1.27 ( $\pm$ 1.37%)	
99.9999%, 4.892 $\sigma_{\bar{x}}$	92.79 $\pm$ 1.407 ( $\pm$ 1.52%)	

Figure 10. Confidence interval of statistical test for different critical value of *t*-test (BraTS 2018 dataset).

#### 4. Discussion

We discuss the results of the proposed method from a critical point of view. The labeled results are illustrated in Figure 6. Three BraTs datasets were used for the validation of the proposed method. The numerical results are presented in Table 1, Table 4, and Table 7. The results presented in these tables were validated through two pipeline procedures, as mentioned in Section 3. The results showed that the accuracy of Pro-FC7 was less, as compared to the proposed architecture. The main reason for the degradation of the classification accuracy was the number of features. For the architecture of VGG19, the feature length on FC7 was 4096, whereas, the feature length of FC8 was 1000; therefore, during the selection process, the target MSER could not be met. Moreover, due to higher number of features, the execution time was also higher for Pro-FC7, as compared to the proposed method.

To give the reader an idea of comparison with the existing techniques, we briefly mentioned some published results. In [24], the authors presented a deep-learning-based system and used the BraTs dataset series for the experimental process. They achieved an accuracy of 97.8%, 96.9%, and 92.5% for the BraTs2015, BraTs2017, and BraTs2018, respectively. Sajjad et al. [23] presented a deep learning model and evaluated this on two datasets—Brain tumor and Radiopaedia. They achieved an accuracy of 94.58% and 90.67%, respectively, on both. Toğaçar et al. [35] achieved an average 96.77% accuracy for the classification of healthy and tumor MRI images. The proposed method achieved an accuracy of 98.16%, 97.26%, and 93.40%, which was better than that compared to the accuracy reported for the state-of-the-art techniques. Additionally, the worst time complexity of our algorithm was  $O(n^3 + k) + C$ , where *k* represents the number of iterations and *C* is a constant term.

In addition, we also calculated the Mathew correlation coefficient (MCC) measure for the ELM classifier; the results are given in Table 13. In this table, it is shown that the MCC values were closer to 1, which showed the better prediction performance of the proposed scheme.

**Table 13.** Calculation of the Mathew correlation coefficient (MCC) value for the ELM classifier.

Dataset	Pro-FC7	Proposed	MCC
BRATS2015	√		0.8690
		√	<b>0.8804</b>
BRATS2017	√		0.8523
		√	<b>0.8764</b>
BRATS2018	√		0.8036
		√	<b>0.8244</b>

The better values are shown in bold.

## 5. Conclusions

This paper presents a fully automated deep learning system, along with contrast enhancement for multimodal brain tumor classification. The strength of this work was in three steps. First, in the preprocessing step, contrast stretching using edge-based texture HE was employed to increase the local contrast of the tumor region. Secondly, the selection of robust deep learning features by implementing correntropy via mutual learning and ELM (CML-ELM) was utilized. Using CML-ELM, the robust features were computed, which were fused through the PLS-based approach, in a later stage. Third, the ELM classifier was implemented for the classification of proposed tumors into the relevant category. The experimental process was conducted on the BraTs datasets and the results showed an improved accuracy (98.16%, 97.26%, and 93.40%, for the BraTs2015, BraTs2017, and BraTs2018 datasets, respectively). The feature selection process was not only helpful for improving the classification accuracy, but also resulted in the reduction of the computational time. Finally, the accuracy results of the proposed method were stable, which could be concluded on the basis of the presented results.

**Author Contributions:** Conceptualization, M.A.K., I.A., and M.A.; methodology, M.A.K., I.A., and M.A.; software, M.A.K., I.A., and M.A.; validation, M.A.K., R.D., and A.R.; formal analysis, M.A.K., R.D., and R.S.; investigation, M.A.K., I.A., M.A., A.R. and S.A.C.B.; resources, M.A.K., I.A., and M.A.; data curation, M.A.K.; writing—original draft preparation, M.A.K., I.A., and M.A.; writing—review and editing, R.D. and R.S.; visualization, M.A.K., I.A., and M.A.; funding acquisition, R.S. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

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

## References

- Deepak, S.; Ameer, P. Brain tumor classification using deep CNN features via transfer learning. *Comput. Biol. Med.* **2019**, *111*, 103345. [CrossRef] [PubMed]
- Sert, E.; Özyurt, F.; Doğantekin, A. A new approach for brain tumor diagnosis system: Single image super resolution based maximum fuzzy entropy segmentation and convolutional neural network. *Med. Hypotheses* **2019**, *133*, 109413. [CrossRef] [PubMed]
- Cancer.Net, Brain Tumor Statistics. 2020. Available online: <https://www.cancer.net/cancer-types/brain-tumor/statistics> (accessed on 24 July 2020).
- Roy, S.; Bandyopadhyay, S.K. Detection and Quantification of Brain Tumor from MRI of Brain and It's Symmetric Analysis. *Int. J. Inf. Commun. Technol. Res.* **2012**, *2*, 477–483.
- Pereira, S.; Pinto, A.; Alves, V.; Silva, C.A. Brain tumor segmentation using convolutional neural networks in MRI images. *IEEE Trans. Med. Imaging* **2016**, *35*, 1240–1251. [CrossRef]
- Tiwari, A.; Srivastava, S.; Pant, M. Brain tumor segmentation and classification from magnetic resonance images: Review of selected methods from 2014 to 2019. *Pattern Recognit. Lett.* **2020**, *131*, 244–260. [CrossRef]
- Hussain, U.N.; Khan, M.A.; Lali, I.U.; Javed, K.; Ashraf, I.; Tariq, J.; Ali, H.; Din, A. A Unified Design of ACO and Skewness based Brain Tumor Segmentation and Classification from MRI Scans. *J. Control Eng. Appl. Inf.* **2020**, *22*, 43–55.

8. Bakas, S.; Reyes, M.; Jakab, A.; Bauer, S.; Rempfler, M.; Crimi, A.; Shinohara, R.T.; Berger, C.; Ha, S.M.; Rozycki, M. Identifying the best machine learning algorithms for brain tumor segmentation, progression assessment, and overall survival prediction in the BRATS challenge. *arXiv* **2018**, arXiv:1811.02629.
9. Chen, L.C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 801–818.
10. Milletari, F.; Navab, N.; Ahmadi, S.A. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In Proceedings of the 2016 Fourth International Conference on 3D Vision (3DV), Stanford, CA, USA, 25–28 October 2016; pp. 565–571.
11. Rashid, M.; Khan, M.A.; Alhaisoni, M.; Wang, S.-H.; Naqvi, S.R.; Rehman, A.; Saba, T. A Sustainable Deep Learning Framework for Object Recognition Using Multi-Layers Deep Features Fusion and Selection. *Sustainability* **2020**, *12*, 5037. [[CrossRef](#)]
12. Khan, M.A.; Khan, M.A.; Ahmed, F.; Mittal, M.; Goyal, L.M.; Hemanth, D.J.; Satapathy, S.C. Gastrointestinal diseases segmentation and classification based on duo-deep architectures. *Pattern Recognit. Lett.* **2020**, *131*, 193–204. [[CrossRef](#)]
13. Majid, A.; Khan, M.A.; Yasmin, M.; Rehman, A.; Yousafzai, A.; Tariq, U. Classification of stomach infections: A paradigm of convolutional neural network along with classical features fusion and selection. *Microsc. Res. Tech.* **2020**, *83*, 562–576. [[CrossRef](#)]
14. Rauf, H.T.; Shoab, U.; Lali, M.I.; Alhaisoni, M.; Irfan, M.N.; Khan, M.A. Particle Swarm Optimization with Probability Sequence for Global Optimization. *IEEE Access* **2020**, *8*, 110535–110549. [[CrossRef](#)]
15. Isensee, F.; Kickingereder, P.; Wick, W.; Bendszus, M.; Maier-Hein, K.H. Brain tumor segmentation and radiomics survival prediction: Contribution to the brats 2017 challenge. In Proceedings of the International MICCAI Brainlesion Workshop, BrainLes 2017, Quebec City, QC, Canada, 14 September 2017; pp. 287–297.
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](#)] [[PubMed](#)]
17. Gumaei, A.; Hassan, M.M.; Hassan, M.R.; Alelaiwi, A.; Fortino, G. A hybrid feature extraction method with regularized extreme learning machine for brain tumor classification. *IEEE Access* **2019**, *7*, 36266–36273. [[CrossRef](#)]
18. Liu, J.; Liu, H.; Tang, Z.; Gui, W.; Ma, T.; Gong, S.; Gao, Q.; Xie, Y.; Niyoyita, J.P. IOUC-3DSFCNN: Segmentation of brain tumors via IOU constraint 3D symmetric full convolution network with multimodal auto-context. *Sci. Rep.* **2020**, *10*, 6256. [[CrossRef](#)] [[PubMed](#)]
19. Ghassemi, N.; Shoeibi, A.; Rouhani, M. Deep neural network with generative adversarial networks pre-training for brain tumor classification based on MR images. *Biomed. Signal. Process. Control* **2020**, *57*, 101678. [[CrossRef](#)]
20. Myronenko, A. 3D MRI brain tumor segmentation using autoencoder regularization. In Proceedings of the International MICCAI Brainlesion Workshop BrainLes 2018, Granada, Spain, 16 September 2018; pp. 311–320.
21. Zhou, C.; Chen, S.; Ding, C.; Tao, D. Learning contextual and attentive information for brain tumor segmentation. In Proceedings of the International MICCAI Brainlesion Workshop BrainLes 2018, Granada, Spain, 16 September 2018; pp. 497–507.
22. Amin, J.; Sharif, M.; Gul, N.; Yasmin, M.; Shad, S.A. Brain tumor classification based on DWT fusion of MRI sequences using convolutional neural network. *Pattern Recognit. Lett.* **2020**, *129*, 115–122. [[CrossRef](#)]
23. Sajjad, M.; Khan, S.; Muhammad, K.; Wu, W.; Ullah, A.; Baik, S.W. Multi-grade brain tumor classification using deep CNN with extensive data augmentation. *J. Comput. Sci.* **2019**, *30*, 174–182. [[CrossRef](#)]
24. Sharif, M.I.; Li, J.P.; Khan, M.A.; Saleem, M.A. Active deep neural network features selection for segmentation and recognition of brain tumors using MRI images. *Pattern Recognit. Lett.* **2020**, *129*, 181–189. [[CrossRef](#)]
25. Seetha, J.; Raja, S.S. Brain tumor classification using convolutional neural networks. *Biomed. Pharmacol. J.* **2018**, *11*, 1457–1461. [[CrossRef](#)]
26. Vijh, S.; Sharma, S.; Gaurav, P. Brain Tumor Segmentation Using OTSU Embedded Adaptive Particle Swarm Optimization Method and Convolutional Neural Network. In *Data Visualization and Knowledge Engineering*; Springer: Cham, Switzerland, 2020; pp. 171–194.
27. Arunkumar, N.; Mohammed, M.A.; Mostafa, S.A.; Ibrahim, D.A.; Rodrigues, J.J.; de Albuquerque, V.H.C. Fully automatic model-based segmentation and classification approach for MRI brain tumor using artificial neural networks. *Concurr. Comput. Pract. Exp.* **2020**, *32*, e4962. [[CrossRef](#)]

28. Sharif, M.; Amin, J.; Raza, M.; Anjum, M.A.; Afzal, H.; Shad, S.A. Brain tumor detection based on extreme learning. *Neural Comput. Appl.* **2020**, *1–13*. [[CrossRef](#)]
29. Ismael, S.A.A.; Mohammed, A.; Hefny, H. An enhanced deep learning approach for brain cancer MRI images classification using residual networks. *Artif. Intel. Med.* **2020**, *102*, 101779. [[CrossRef](#)] [[PubMed](#)]
30. Sharif, M.; Tanvir, U.; Munir, E.U.; Khan, M.A.; Yasmin, M. Brain tumor segmentation and classification by improved binomial thresholding and multi-features selection. *J. Amb. Intel. Hum. Comp.* **2018**, *1–20*. [[CrossRef](#)]
31. Khan, M.A.; Lali, I.U.; Rehman, A.; Ishaq, M.; Sharif, M.; Saba, T.; Zahoor, S.; Akram, T. Brain tumor detection and classification: A framework of marker-based watershed algorithm and multilevel priority features selection. *Microsc. Res. Tech.* **2019**, *82*, 909–922. [[CrossRef](#)] [[PubMed](#)]
32. Ke, Q.; Zhang, J.; Wei, W.; Damaševičius, R.; Woźniak, M. Adaptive independent subspace analysis of brain magnetic resonance imaging data. *IEEE Access* **2019**, *7*, 12252–12261. [[CrossRef](#)]
33. Rehman, A.; Naz, S.; Razzak, M.I.; Akram, F.; Imran, M. A deep learning-based framework for automatic brain tumors classification using transfer learning. *Circ. Syst. Signal. Pr.* **2020**, *39*, 757–775. [[CrossRef](#)]
34. Ghosal, P.; Nandanwar, L.; Kanchan, S.; Bhadra, A.; Chakraborty, J.; Nandi, D. Brain Tumor Classification Using ResNet-101 Based Squeeze and Excitation Deep Neural Network. In Proceedings of the Second International Conference on Advanced Computational and Communication Paradigms (ICACCP), Gangtok, India, 25–28 February 2019; pp. 1–6.
35. Toğaçar, M.; Cömert, Z.; Ergen, B. Classification of brain MRI using hyper column technique with convolutional neural network and feature selection method. *Expert Syst. Appl.* **2020**, *149*, 113274. [[CrossRef](#)]
36. Muhammad, K.; Khan, S.; Del Ser, J.; de Albuquerque, V.H.C. Deep Learning for Multigrade Brain Tumor Classification in Smart Healthcare Systems: A Prospective Survey. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**. [[CrossRef](#)]
37. Rehman, A.; Khan, M.A.; Mehmood, Z.; Saba, T.; Sardaraz, M.; Rashid, M. Microscopic melanoma detection and classification: A framework of pixel-based fusion and multilevel features reduction. *Microsc. Res. Tech.* **2020**, *83*, 410–423. [[CrossRef](#)]
38. Khan, M.A.; Sharif, M.; Akram, T.; Bukhari, S.A.C.; Nayak, R.S. Developed Newton-Raphson based deep features selection framework for skin lesion recognition. *Pattern Recognit. Lett.* **2020**, *129*, 293–303. [[CrossRef](#)]
39. Gabryel, M.; Damaševičius, R. The image classification with different types of image features. In Proceedings of the International Conference on Artificial Intelligence and Soft Computing, ICAISC 2017, Zakopane, Poland, 11–15 June 2017; pp. 497–506. [[CrossRef](#)]
40. Khan, M.A.; Rubab, S.; Kashif, A.; Sharif, M.I.; Muhammad, N.; Shah, J.H.; Zhang, Y.-D.; Satapathy, S.C. Lungs cancer classification from CT images: An integrated design of contrast based classical features fusion and selection. *Pattern Recognit. Lett.* **2020**, *129*, 77–85. [[CrossRef](#)]
41. Chouhan, V.; Singh, S.K.; Khamparia, A.; Gupta, D.; Tiwari, P.; Moreira, C.; Damaševičius, R.; de Albuquerque, V.H.C. A novel transfer learning based approach for pneumonia detection in chest X-ray images. *Appl. Sci.* **2020**, *10*, 559. [[CrossRef](#)]
42. Ke, Q.; Zhang, J.; Wei, W.; Połap, D.; Woźniak, M.; Kośmider, L.; Damaševičius, R. A neuro-heuristic approach for recognition of lung diseases from X-ray images. *Expert Syst. Appl.* **2019**, *126*, 218–232. [[CrossRef](#)]
43. Weiss, K.; Khoshgoftaar, T.M.; Wang, D. A survey of transfer learning. *J. Big Data* **2016**, *3*, 9. [[CrossRef](#)]
44. Huang, G.-B.; Zhu, Q.-Y.; Siew, C.-K. Extreme learning machine: A new learning scheme of feedforward neural networks. In Proceedings of the 2004 IEEE International Joint Conference on Neural Networks, Budapest, Hungary, 25–29 July 2004; pp. 985–990.

