Mathematical Biosciences and Engineering

MBE, 18(2): 1550–1572.
DOI: 10.3934/mbe.2021080
Received: 27 November 2020
Accepted: 25 January 2021
Published: 29 January 2021

http://www.aimspress.com/journal/MBE

Research article

A comparative study for glioma classification using deep convolutional neural networks

Hakan Özcan[1,] *, Bülent Gürsel Emiroğlu[2], Hakan Sabuncuoğlu[3], Selçuk Özdoğan[4], Ahmet Soyer[3] and Tahsin Saygı[5]

[1] Department of Computer Technology, Amasya University, Amasya, Turkey
[2] Department of Computer Engineering, Kırıkkale University, Kırıkkale, Turkey
[3] Department of Neurosurgery, Ufuk University, Ankara, Turkey
[4] Adatıp Hospital, Neurosurgery Clinic, İstanbul, Turkey
[5] Department of Neurosurgery, Haseki Research and Training Hospital, İstanbul, Turkey

* **Correspondence:** Email: hozcan@amasya.edu.tr; Tel: +90-358-211-5005.

Abstract: Gliomas are a type of central nervous system (CNS) tumor that accounts for the most of malignant brain tumors. The World Health Organization (WHO) divides gliomas into four grades based on the degree of malignancy. Gliomas of grades I-II are considered low-grade gliomas (LGGs), whereas gliomas of grades III-IV are termed high-grade gliomas (HGGs). Accurate classification of HGGs and LGGs prior to malignant transformation plays a crucial role in treatment planning. Magnetic resonance imaging (MRI) is the cornerstone for glioma diagnosis. However, examination of MRI data is a time-consuming process and error prone due to human intervention. In this study we introduced a custom convolutional neural network (CNN) based deep learning model trained from scratch and compared the performance with pretrained AlexNet, GoogLeNet and SqueezeNet through transfer learning for an effective glioma grade prediction. We trained and tested the models based on pathology-proven 104 clinical cases with glioma (50 LGGs, 54 HGGs). A combination of data augmentation techniques was used to expand the training data. Five-fold cross-validation was applied to evaluate the performance of each model. We compared the models in terms of averaged values of sensitivity, specificity, F1 score, accuracy, and area under the receiver operating characteristic curve (AUC). According to the experimental results, our custom-design deep CNN model achieved comparable or even better performance than the pretrained models. Sensitivity, specificity, F1 score, accuracy and AUC values of the custom model were 0.980, 0.963, 0.970, 0.971 and 0.989, respectively. GoogLeNet showed better performance than AlexNet and SqueezeNet in terms of accuracy and AUC with a sensitivity, specificity, F1 score, accuracy, and AUC values of

0.980, 0.889, 0.933, 0.933, and 0.987, respectively. AlexNet yielded a sensitivity, specificity, F1 score, accuracy, and AUC values of 0.940, 0.907, 0.922, 0.923 and 0.970, respectively. As for SqueezeNet, the sensitivity, specificity, F1 score, accuracy, and AUC values were 0.920, 0.870, 0.893, 0.894, and 0.975, respectively. The results have shown the effectiveness and robustness of the proposed custom model in classifying gliomas into LGG and HGG. The findings suggest that the deep CNNs and transfer learning approaches can be very useful to solve classification problems in the medical domain.

Keywords: glioma; clinical scans; retrospective study; magnetic resonance imaging; classification; deep convolutional neural networks; transfer learning

1. Introduction

A glioma is a most common type of central nervous system (CNS) tumor that originates in the brain or spinal cord [1]. Gliomas comprise about 80% of all malignant brain tumors and make up 30% of all the CNS tumors [2]. Every year in the United States, around 10,000 people are diagnosed with malignant glioma, and in about only 25% of the cases, patients survive a year following the diagnosis [3]. According to the updated World Health Organization (WHO) classification schema, gliomas are grouped into four grades ranging from I to IV, that are based on the type of malignancy analyzed through the histological features and genetic characteristics of glioma tissue [4]. Grades I and II gliomas are defined as "low grade gliomas" (LGGs), whereas gliomas of grades III and IV are considered "high grade gliomas" (HGGs) [5]. HGGs grow rapidly and aggressively in an infiltrative manner [6]. Even with current treatment options, such as chemotherapy, radiation and surgery, the prognosis of HGGs is still very poor, and patients with HGG are rarely cured completely [7]. The median survival time with the most common HGG, termed glioblastoma, is less than 15 months [8]. LGGs, on the other hand, grow slower than HGGs and their median survival ranges from 5 to 10 years. [9]. Although LGGs are often found to have a better prognosis than HGGs, they can show anaplastic progression (i.e., malignant transformation) to HGGs over time [10]. Therefore, diagnostic accuracy of detecting LGGs before the malignant transformation to HGGs plays a critical role in optimal treatment planning.

According to current clinical practice, biopsy still matters in a histological glioma diagnosis and is considered the definitive standard for glioma grading and treatment planning [11]. However, obtaining a biopsy from a brain involves invasive procedures that are not only uncomfortable for patients but also comprise potential risk of complications, such as bleeding, infection, stroke, and seizures [12]. Besides that, in some cases, especially in occurrence of malignant progression, further biopsies can be required to guide treatment decisions [13], hence the potential complications can occur each time a follow-up biopsy is taken. Even though primary diagnosis of glioma may still require a tissue specimen taken by biopsy, non-invasive approaches help to avoid unnecessary biopsies by using various imaging methods, especially Magnetic Resonance Imaging (MRI) [14].

MRI is one of the most common non-invasive diagnostic methods and has been shown to be very sensitive in evaluating soft tissue masses without exposing radiation and become a valuable tool for clinical decision-making prior to or along with biopsy [15]. In fact, MRI is useful for detection and classification of glioma grades as it offers a broad range of imaging sequences with relevant

protocols, such as T1-weighted (T1W), T2-weighted (T2W) and fluid-attenuated inversion-recovery (FLAIR) [16]. However, examination of MRI data with a large series of images is a time-consuming process, and interpretations of tissue characteristics and patterns on images are subject to human perception and may vary among experts [17]. Such variability in interpretations of MRI images, indeed, can potentially result in biases and lead to classification errors of false-positive or false-negative diagnosis [18].

Many efforts have been made on glioma grade diagnosis using traditional machine learning methods [19–24]. In these methods, researchers have used histogram and texture analysis and benefited from statistical techniques such as logistic regression, random forest, and support vector machines in the classification process. Although these efforts contributed to analysis of glioma grades to some extent, they involved complex procedures of extracting hand-crafted features in an observer-dependent way. But thanks to recent advances in the field of artificial intelligence, new algorithms were provided for addressing this complexity. When these algorithms are basically built on multi-layer neural networks, they are characterized as Deep Learning. Deep learning, as a promising sub-branch of machine learning, can learn from sample data and provide the ability to address the challenges in many classification problems [25]. A convolutional neural network (CNN) [26] is a feed-forward neural network used in deep learning. CNNs require less preprocessing of data than other algorithms do [27]. This makes them more advantageous than the conventional methods in terms of reducing human-intervention and creating more automated learning models.

When building a typical CNN model, using an optimal optimization algorithm to explore the best matching weights and biases for the input and output data is called as training. Two main approaches can be used in the development of CNN-based models. The first approach is that the network is originally designed and trained from scratch. The second approach, on the other hand, is utilizing the pretrained networks and is called Transfer Learning. Designing a network and training from scratch require well-labeled data and involve a computationally expensive process. In order to alleviate computational burden and mitigate the limitation of small data size, transfer learning methods with pre-trained CNNs previously trained on large image datasets can be used [28].

In this study we proposed a new custom-design CNN-based deep learning model (Model 1) trained from scratch and compared the performance with the pretrained networks of AlexNet (Model 2), GoogLeNet (Model 3) and SqueezeNet (Model 4) through transfer learning for an effective glioma grade prediction. The focus of the study also involves how CNN-based deep learning methodologies can be applied in a real-world clinical environment where only limited data are available. For this purpose, we conducted a retrospective review of MRI data of 104 patients with the diagnosis of glioma between December of 2016 and October 2019. Then, we trained and tested the models with the data to facilitate the classification of low- and high-grade glioma. The major contributions of this research are listed as follows: 1) A custom-design deep CNN architecture was proposed for 2-dimensional (2D) MRI images for an effective glioma grade prediction. 2) To the best of our knowledge, this is the first work that compare the glioma grade prediction performance of SqueezeNet through transfer learning with a comparison to other commonly applied pretrained networks of AlexNet and GoogLeNet. 3) A new glioma dataset was retrospectively curated; learned features, morphological patterns were analyzed in the light of the four deep learning models.

The rest of this paper is organized as follows: Section 2 presents a literature review of the machine learning approaches recently applied to glioma diagnosis. Section 3 describes the materials and methods that we used to develop the CNN models and transfer learning workflow. Section 4

depicts experimental results of the models. Finally, Section 5 summarizes the conclusions and discusses the results.

2. Related work

Previous studies on glioma analysis have focused on either classical machine learning methods or, to a lesser extent, deep learning approaches. For example, Zacharaki et al. [29] proposed a machine learning schema based on radiomic features for distinguishing glioma types. They applied classification models based on linear discriminant analysis, k-nearest neighbors, and support vector machine (SVM) algorithms with the extracted features such as tumor shape, intensity, and texture features to quantitatively evaluate gliomas. Their SVM classifier performed better, achieving an accuracy of 88%. Kang at al. [19] applied histogram analysis of apparent diffusion coefficient maps for differentiating astrocytic tumors from grade II-IV gliomas and achieved an 80% classification accuracy. Ditmer et al. [30] implemented texture analysis with a filtration- histogram based approach using radiomic features on T1, FLAIR, and diffusion weighted imaging (DWI) MRI sequences to differentiate LGGs and HGGs. This work showed that T1-weighted and FLAIR images provided better results compared to DWI-generated ADC maps and attained an AUC of 90%. Banerjee et al. [31] explored the potential use of deep learning-based approaches in grading gliomas from MR images. They tested the suitability of transfer learning by applying VGGNet and ResNet architectures, an attained accuracy of 84%×90% for 2D images. In another study [32], researchers presented two glioma grading methods, which include brain tumor segmentation using a modified U-Net model. They used a regional convolutional neural network (R-CNN) for the classification task in each of the 2D image slices of MRIs. Their proposed 2D Mask R-CNN yielded an accuracy of 96%. The classification performance of the 2D model showed that data augmentation improved the results [32]. In a recent review [33], it has been stated that machine learning methods give insufficient information about how tissues are classified in glioma analysis, however, more explanatory information can be presented by using visualization methods within CNN-based deep learning models. A report [34] underlining the effect of using deep learning on glioma diagnoses summarizes that deep learning algorithms can easily be adopted in clinical practice and different data sets would be useful in investigating the factors in the classification of gliomas.

3. Materials and methods

3.1. Database

The study flow included data acquisition, preprocessing, data splitting, data augmentation, classification, and evaluation phases as shown in Figure 1. In the data acquisition phase, clinical MRI data of 104 cases including 50 with LGG (grade II) and 54 with HGG (grade IV) were retrospectively recruited for this study. Data were extracted from anonymized MRI scans, which were performed in head-first, supine orientation, using 1.5T Philips Achieva MRI scanner (A-series, USA). MRI protocols mainly consisted of T1W, T2W, FLAIR and DWI sequences recorded in The Digital Imaging and Communications in Medicine (DICOM) format in the sagittal, coronal, and axial planes of the brain. The DICOM images were in nested sequences in gray scale and rendered with contrast-based intensities for tumor regions. After our evaluations of the sequences in terms of the slice quality and quantity of tissues, only axial T2W/FLAIR (N = 104) cases, on which glioma

lesions appear hyperintense with 5–8mm slices from fluid-attenuated MRI, were included in this study. The final diagnosis for each case was analyzed based on the pathology reports and each MRI sequence was stratified into LGG and HGG in accordance with the WHO criteria by a group of experts from neurosurgery and neuroradiology. The retrospective review consisted of the cases between December of 2016 and October 2019 as listed in Table 1.

Table 1. Characteristics and distribution of the glioma cases used in the study across years.

Year	HGG	LGG	T2W/FLAIR
2016	4	3	7
2017	12	21	33
2018	21	10	31
2019	17	16	33
Total	54	50	104

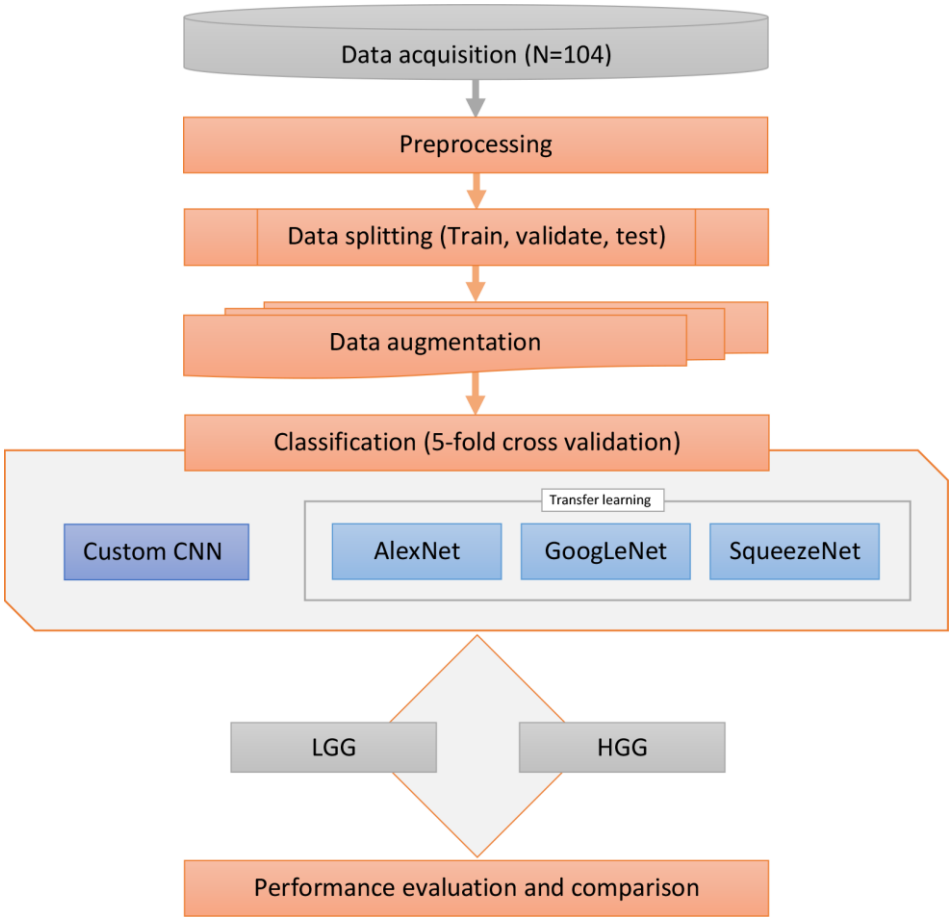

Figure 1. The design of the study.

3.1.1. Ethics approval of research

Ethical approval (15386878-044/2020) was obtained from the Institutional Ethical Committee of Amasya University, and all procedures were performed in line with the tenets of the Declaration of Helsinki.

3.2. *Preprocessing*

In the preprocessing, we applied a user-guided segmentation for the delineation of region of interest (ROI) in relation to the glioma tissues being evaluated. ROIs were outlined in the MRI slices with the largest tumor cross-section and centered on the corresponding tissues. We performed multiple cropping and obtained ~5 images at each set of slices. This helped expand the data, without actual introducing new data [35,36]. Then we saved ROI images in BMP format with the dimensions of 64×64px. Finally, we obtained 518 images. After that, the raw pixel values in each image were normalized to have a mean of zero and a standard deviation of 1. Channel-wise standardization and z-score normalization were also applied during the convolution and pooling operations to facilitate the training processes.

3.3. *Data splitting*

Cross validation is used especially in studies with small datasets to provide generalizability of models in machine learning. In this study, we split the image dataset into train, validate and test sets in a nested cross validation manner as shown in Figure 2. K-fold cross validation method was used to estimate the out of sample classification accuracy of the models. Balanced random sampling strategy was used to ensure each fold has a balanced representation of classes. We divided the dataset into five subsets (folds) and evaluated the results by experimenting on each subset. One of the folds (20%) is used for testing, the remaining four folds (80%) are used to train the dataset and this process is repeated five times for each model. At each cross-validation iteration, the model was trained using a training/validation set and evaluated on a separate testing set in a patient-based manner. Finally, average performance results were calculated to compare the models.

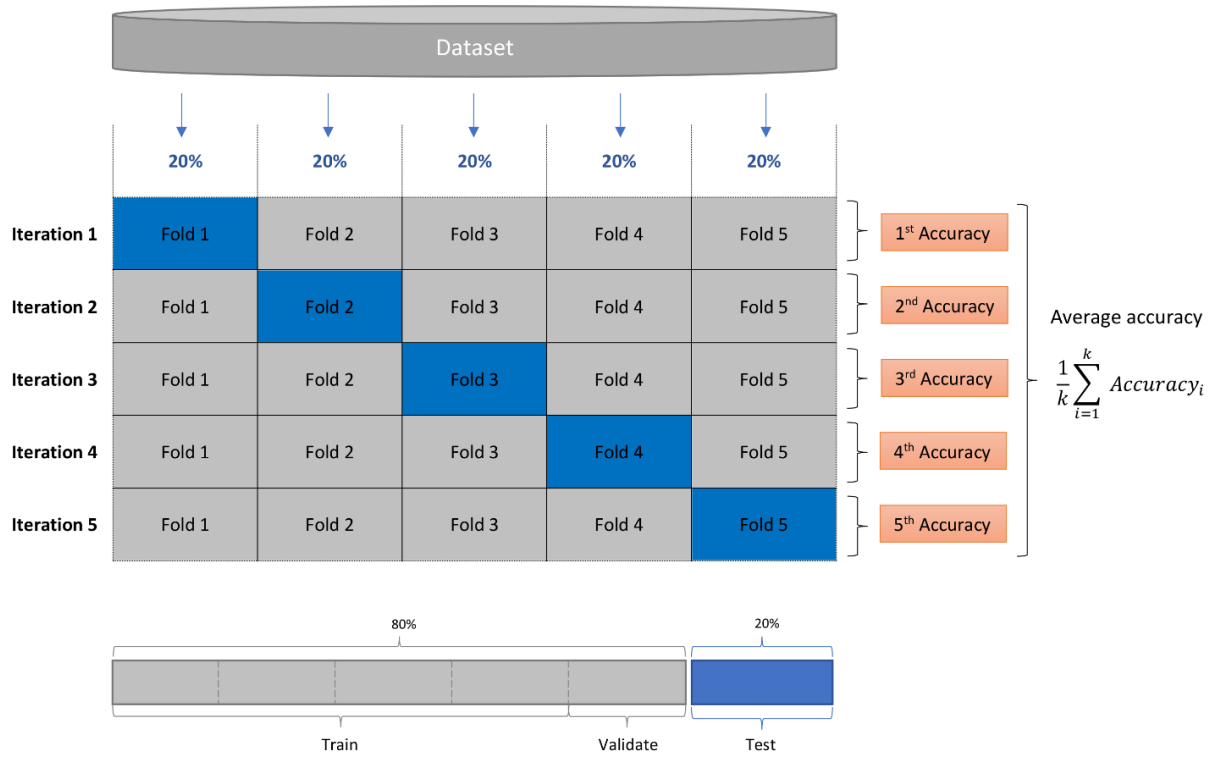

Figure 2. Data splitting in a five-fold cross validation manner.

3.4. Data augmentation

Data augmentation is a regularization strategy that artificially expands the volume and diversity of data. One of the common problems in deep learning-based applications in medical image analysis is the lack of representative data [37]. There are many data augmentation techniques available addressing this issue such as rigid transformations including flipping, rotating, and translating. We applied random combinations of these techniques to increment the dataset. We made use of the Augmenter [38] library, which uses a stochastic, pipeline-based approach to expand training dataset. By using this library, we chained several augmentation techniques together as listed in Table 2. We applied probability-based random rotation, zooming, shearing, and flipping to our training set. Additionally, we used random elastic gaussian transforms, which enabled grid-based distortions on ROIs. We preserved the ROI focus area not to lose the largest part of the lesion. Finally, the training data were increased to 20-fold larger in size. A set of augmented images are shown in Figure 3.

Table 2. Data augmentation techniques with the probabilities.

Augmentation techniques	Probability
Rotating and zooming	0.20
Distortion and skewing	0.30
Shearing and flipping	0.50

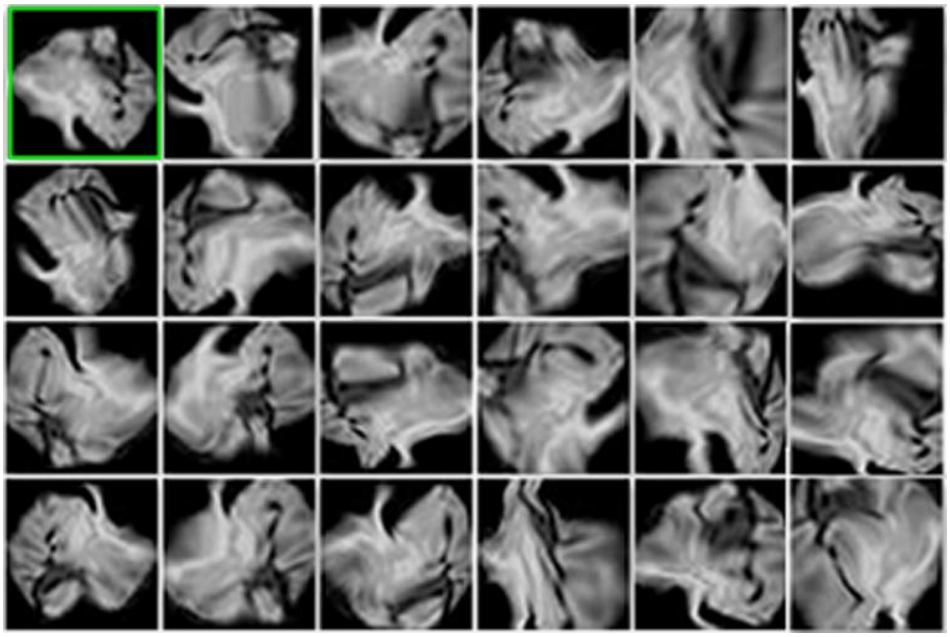

Figure 3. Representative examples of augmented images: The image located at top-left is the original image.

3.5. The CNN architecture

We proposed a custom deep CNN architecture to train from scratch. The architecture consists of 7 linearly structured convolutional layers. Each convolutional layer applies a set of filters (kernels) to

its preceding layer with a stride of 1 to extract feature maps (activation maps), such as edges, lines, corners, and gradients. The first two convolutions begin with 3×3 filters to obtain high-level features, such as texture and shapes. After that, a 5×5 filter is applied to the next three layers. For the sixth and seventh convolutions, filters with sizes of 7×7 and 3×3 are implemented, respectively. All the features obtained from each convolutional layer are flattened and fed into a fully connected layer. Finally, the classification is performed through the fully connected layer using the SoftMax activation as a two-class output with the cross-entropy loss function. Figure 4 depicts the overall architecture of the CNN model.

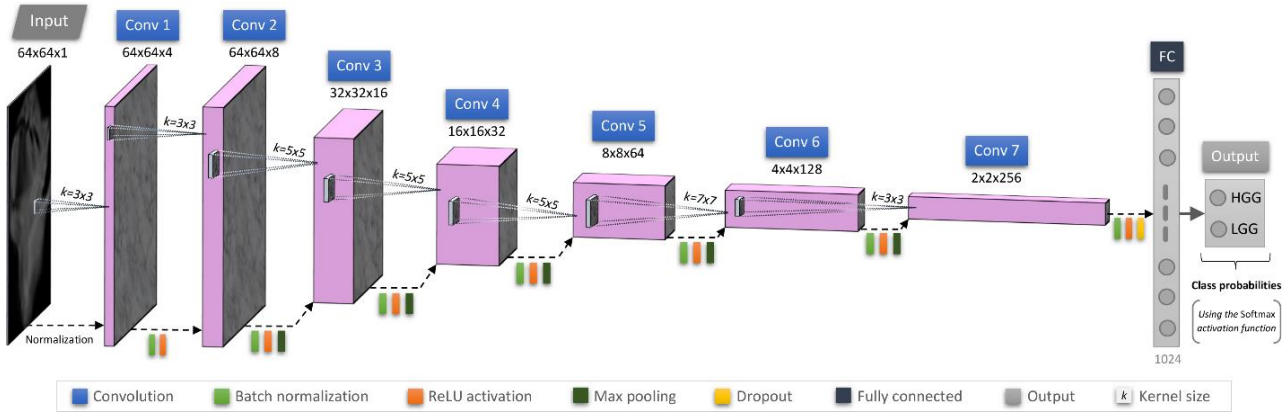

Figure 4. Proposed CNN architecture.

Feature maps are formed as a product of inputs and kernels. Both kernels, which hold the learned weights, and inputs fed into a convolutional layer are multidimensional arrays. So, the convolution operation can be expressed as a pair-wise product of two 2-dimensional matrices, given by

$$F[i,j] = K[i,j] \otimes I[i,j] = \sum_{m=-k}^{k} \sum_{n=-k}^{k} I[m,n] \bullet K[i-m, j-n] \tag{1}$$

where F represents the calculated feature map, K is the kernel, and I is the input values. After applying convolution to input I with the kernel K of width $2k$, each feature map forms a 2-dimensional matrix of features.

We preferred odd number kernel sizes with zero-padding strategy [39] to maintain spatial dimensionality of input images. To facilitate the convergence of the network especially for a high learning rate, we applied batch normalization method [40] to each convolution layer. In this way, we aimed to stabilize distribution of inputs before each activation.

$$BN(x_i) = \gamma_i \bar{x}_i + \beta_i \tag{2}$$

In Eq 2, gamma (γ) indicates the scaling factor, and beta (β) is the shifting parameter for the normalized weights. Both β and γ are learnable parameters within the network. $\bar{x}_i$ represents a normalized output value of the previous layer. Batch normalization multiplies the normalized values by the standard deviation and adds the mean value. This procedure helps increase the stability of the model and reduces the overfitting problem [41].

After each batch normalization procedure, rectified linear unit (ReLU) [42] is used in convolutional layers. The contribution of ReLU to accelerating the converging speed of CNNs has been empirically proven and it is widely used as the activation function in deep models [43,44]. ReLU is a piecewise linear function that normalizes the output of each neuron. This activation function assigns zero to values below zero or directly outputs input value if it is greater than or equal to zero as shown in Figure 5.

$$\sigma = \max(0, x_i) \in [0, \infty] \tag{3}$$

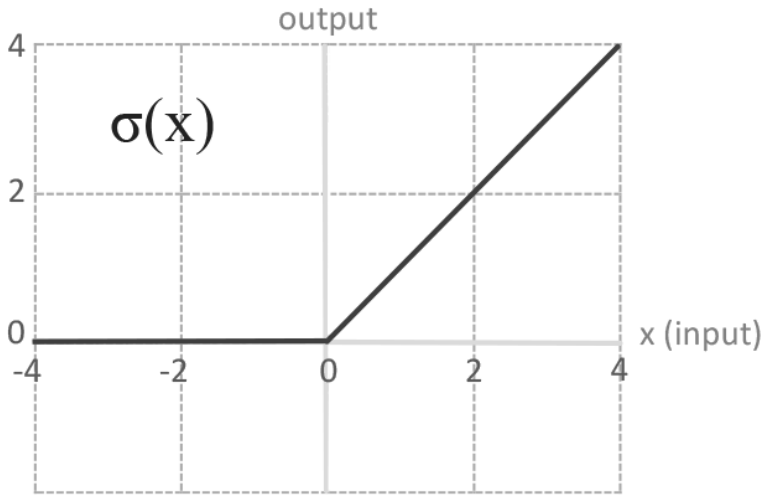

Figure 5. ReLU activation.

We used the max-pooling process [45] after each ReLU activation layer except the last one. This process reduces the spatial size of inputs as it downsamples the features to detect contours, sharp edges, and corners etc. All the max-pooling operations are applied symmetrically using a stride of 2 with zero padding to distribute the pooling procedure over the whole input image. Max polling can be given by

$$h_{mn}^{a} = max_{i=0,\ldots,s,j=0,\ldots,s} h_{(m+i)(n+j)}^{a-1} \tag{4}$$

where s is the shift size of the pooling window over the pixels, and h represents pooling output as the activation of the layer. A pooling window moves over the input with directions in the n^{th} row and m^{th} column by the iteration indices i and j. At each point where the pooling window is located, the maximum value is calculated and recorded on the pooling region. This process divides the input into sub regions and yields a type of discretization of inputs.

Before combining the features maps, a dropout regularization [46] is added to the architecture (p = 0.5). This method randomly sets activation values to zero with a given rate to reduce overfitting of the model. Recall that, the activation function in Eq 1. obtains the sum of input values multiplied by weights in the kernel matrix. The application of dropout to this summation with a certain probability can be formulated as follows:

$$y_i = w_i * \delta + b_i \tag{5}$$

where lambda (δ) is either 1 with the probability of *p,* or 0 otherwise. So, the weights are removed at each step during the element-wise multiplication if lambda is 0 or preserved if lambda is 1. b represents the bias value for the i^{th} feature map.

Feature maps are flattened and connected to a fully connected layer at the end of the architecture. The fully connected layer is then followed by a 2-class output layer, where we appended the Softmax activation function to calculate the probabilities of each class. So, the output layer returns the class having the highest probability. The Softmax function can be formulated as follows:

$$f_i(x) = \frac{e^{x_i}}{\sum_{k=1}^{n} e^{x_k}} \tag{6}$$

where x is the given input vector, transformed into the standard exponential function e^{x_i}, then normalized across all the values so that the sum becomes 1. In other words, while the initial values that the model outputs are in non-normalized form, Softmax converts them into normalized probabilities as an output vector, i.e., probability distribution for n number of classes.

In the experiments, we used stochastic gradient descent (SGD) optimization algorithm [47] with the cross-entropy (CE) loss function. SGD gives a way of effectively minimizing the loss function. As we have a classification problem with two classes LGG and HGG, we used binary cross-entropy as the loss function. The loss function (also known as cost function) calculates how well the CNN algorithm performs over new data by providing a difference between our prediction and the actual observation (ground truth). The binary cross-entropy loss function is formulated as follows:

$$L(f(x), y) = -\frac{1}{n}\sum_{i}^{n} y_i \log\big(f_i(x)\big) + (1 - y_i) \cdot \log\big(1 - f_i(x)\big) \tag{7}$$

where the L function expects two parameters to compute the cross entropy. The first parameter is the output of $f(x)$, which is the Softmax output, i.e., probability distribution of the predicted values as detailed in Eq 6. The second parameter is y, which represents the true class label in the target vector with n samples.

3.6. Transfer learning

Deep CNNs pretrained on a large amount of real-world images offer the possibility of transfer learning implementations tailored to other domain-specific problems [25]. There are many state-of-the-art CNN architectures that have proven their potential in the medical domain [27]. These architectures are typically pre-trained and benchmarked based on the ImageNet [48], a large-scale dataset, which consists of millions of images divided into thousands of object categories. In our study, adaptation of pre-trained CNN architectures involved changing the weights in initial layers and fine-tuning (retraining) some of the last layers with our dataset and optimizing the models' hyper-parameters such as learning rate, mini-bath sizes, and number of training epochs. We implemented transfer learning by using the deep CNNs including AlexNet, GoogLeNet and SqueezeNet, as shown in Figure 6. Then, we compared the performance of the models in predicting LGGs and HGGs.

AlexNet [49] is the first deep CNN architecture that achieved great classification accuracy on ImageNet database in 2012. The structure basically contains eight consecutive layers, which consists of five convolutions followed by three fully connected layers. The model combines three

convolutional layers with max-pooling techniques for the feature maps. ReLU is used as the activation function after each convolution. Dropout regularization is applied to the first two fully connected layers. The input to the AlexNet is with a fixed size of 227×227 RGB image (i.e., 227× 227×3). The output is generated through a 1000-class Softmax classifier built in the last fully connected layer. AlexNet is trained on the ImageNet database and it holds nearly 61 million trainable parameters [49]. On the other hand, GoogLeNet [50], also known as Inception v1, introduced a new way of stacking structure with a unique component to the deep CNN architectures, named, "inception module". The model has 22 layers and is based on the hierarchical combination of inception modules. Each inception module represents a separate block with its own six convolutional layers as well as a pooling layer. In the network architecture, a 1×1 convolution is used to reduce dimensionality of feature vectors (i.e., dimensionality reduction) for faster computations. Inception layers in each block can use multiple kernels of different sizes, particularly of 1×1, 3×3, and 5×5. The architecture uses global average pooling for the purpose of fully connected layers and accepts 224×224×3 images, as an input by default. GoogLeNet is also trained on the ImageNet database and it contains roughly 7 million trainable parameters [50]. As for SqueezeNet [51], it was introduced as a simplified computer vision model in 2016. The model has an 18-layer deep architecture, which is initialized with a standalone convolutional layer and ends with a Softmax output via global average pooling technique. The architecture includes three max pooling layers and four pairs of building blocks, namely fire modules 2–9. A fire module consists of a squeeze layer with two branching expand layers using kernel sizes of 1x1 and 3x3 as shown in Figure 6. These modules are essential parts of the architecture and their main purpose is to reduce the number of parameters within the network. SqueezeNet holds about 1.2 million parameters and expects input size of 227×227×3 [51]. The comparison of the deep CNNs used in this study is summarized in Table 3.

Table 3. Comparison of AlexNet, GoogLeNet and SqueezeNet.

Name	Year	Input size	Depth	Modules	Parameters	Output class
AlexNet	2012	227×227×3	8	N/A	61M	1000
GoogLeNet	2014	224×224×3	22	Inception	7M	1000
SqueezeNet	2016	227×227×3	18	Fire	1.2M	1000

We followed three main steps to apply transfer learning workflow to the pretrained networks. First, we loaded and analyzed each network architecture. Then we replaced the last few layers with custom layers to match the number of classes in our classification problem. Finally, we retrained the networks on our dataset. For all the pretrained networks, we increased the learning rate of the new layers by a constant factor (20%) for the weights and biases. This provided more frequent updates for learning features during the training. As for the initial layers, we preserved the defined learning rate, so they used global learning rate given in Table 4. Weights of the initial layers were updated during the training process. Figure 6 shows the layers of the architectures based on the transfer learning workflow. More specifically, for the AlexNet, we removed the last three layers. We added a new fully connected layer (activations: 1×1×2, weights: 2×4096, bias: 2×1) in place of the layer, named "fc8" (1×1×1000 activations, 1000×4096 weights, 1000×1 bias). Then we appended a Softmax layer (1x1x2) and a classification layer using cross entropy loss function with classes "LGG" and "HGG" at the end of the architecture. For the GoogLeNet, after extracting the layer graph, we replaced the fully connected layer (1×1000 activations, 1000×1024 weights, 1000×1 bias), named

"loss3-classifier", with a new fully connected layer (1×1×2 activations, 2×1024 weights, 2×1 bias). Then the Softmax layer, named “prob” and the classification layer, named “output”, were replaced with a two-class SoftMax classifier and an output with a cross-entropy loss, respectively. As for the SqueezeNet, a convolution layer with stride (1,1) was inserted in place of the convolution layer named "conv 10". Therefore, the ReLU layer (“relu_conv10”) activations were changed from 14×14×1000 to 14×14×2 and the global average pooling layer (“pool10”) activations were set as 1×1×2. The Softmax layer (“prob”) and classification output layer ("classification layer predictions") were replaced in the same way as we did in the AlexNet and GoogLeNet.

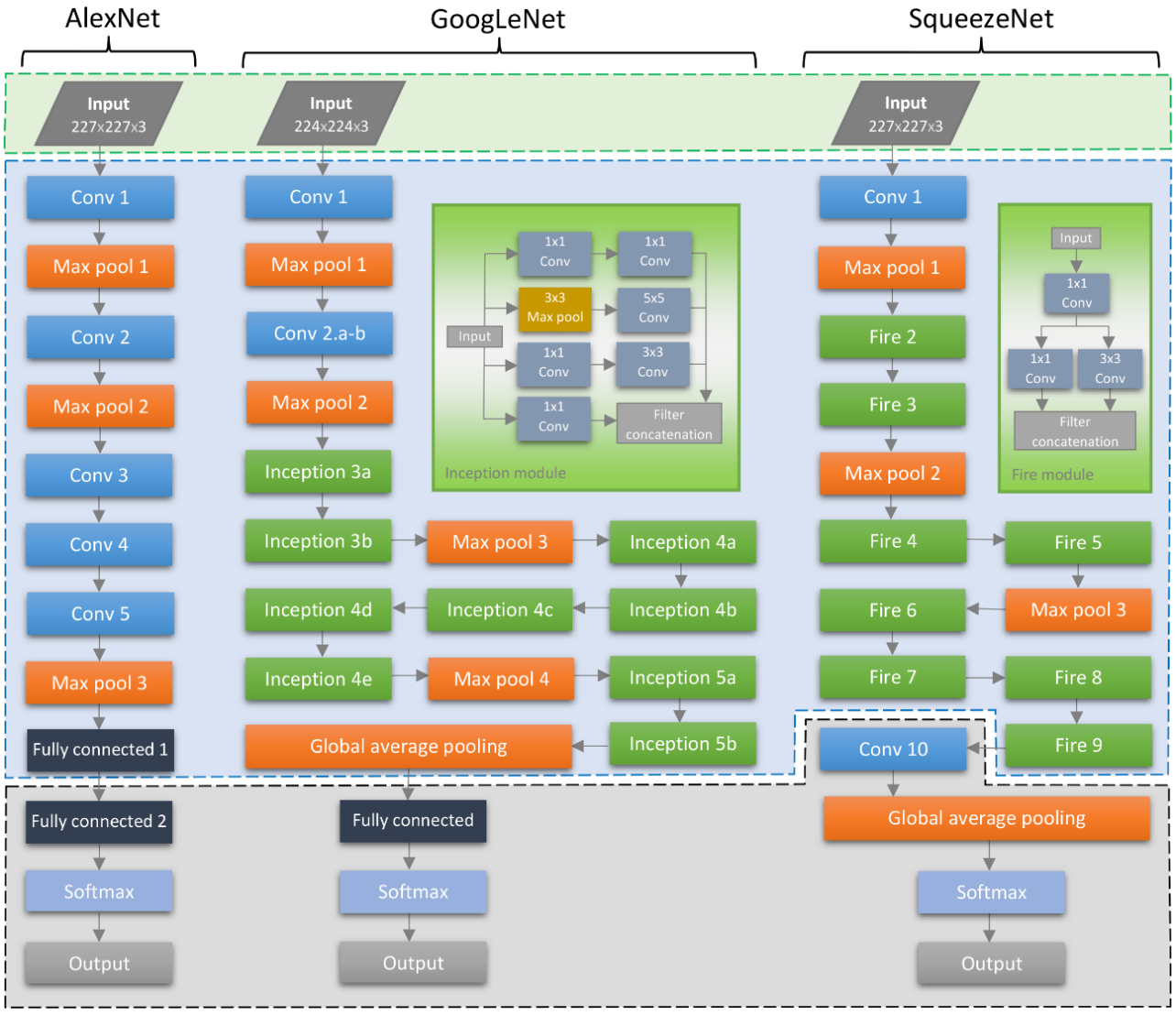

Figure 6. AlexNet, GoogLeNet and SqueezeNet architectures, where the modified/replaced layers based on the transfer learning workflow are enclosed by the dashed black lines; the initial layers, whose weights are updated during the retraining process, are enclosed by the dashed blue lines; and the input layers, according to which the input images are resized during the retraining and testing phases, are enclosed by the dashed green lines.

3.7. Experimental setup

The classification algorithms were developed and tested in MATLAB 2020a on Dell Precision T7810 Tower with the specifications of 64 GB RAM and Intel Xeon E5-2637 quad-core processor. In the experiments, we used the SGD optimizer with an initial learning rate of 0.001. As our images in the dataset were 2D arrays, we converted them to 3-dimensional matrices by repeating the original image array three times on a new dimension to suit the needs of pretrained networks. Then we resized images to 224×224×3 for GoogLeNet and 227×227×3 for AlexNet and SqueezeNet. During the training process of all models, we optimized hyper parameters. After several pre-experimental training phases, the hyper-parameters were set according to optimal performance found for each model. We used a mini batch size of 128, with which we had optimal learning rates. We performed random shuffling of the data for each epoch. During the training of the models, early stopping technique [52,53] was used as it provided the ability to stop the training after a specific number of epochs (defined by "validation patience" in Table 4) when the validation error starts increasing, and the ridge regression was set with a 0.0001 L2 regularization factor. The hyper-parameters used in training are summarized in Table 4.

Table 4. Hyper-parameters.

Parameter	Value
Initial learning rate	1.00e-03
Validation frequency	10
Mini batch size	128
Gradient threshold method	L2 norm
Gradient threshold	Inf
Validation patience	8
Maximum number of epochs	40
Shuffle	Every epoch
L2 regularization factor	1.00e-04
Momentum	0.9

3.8. Evaluation metrics

To measure and compare classification performance of the models, we used the confusion matrix-based metrics, such as sensitivity, specificity, precision, F1 score and accuracy.

A confusion matrix is a table that reports the number of correct and incorrect estimates of the classification model over a given dataset. In our study, the classification problem is binary in nature, so we have two classes, i.e., LGG and HGG. Positive classes are referred to the LGG cases, whereas negative classes are characterized as the HGG cases. True positives (TP) is the number of LGG cases that the model correctly predicted. True negatives (TN) represents the number of HGG cases that are correctly identified by the model. False Positives (FP) is the number of cases that are incorrectly predicted as LGG, although they are HGG. False negatives (FN) is the number of LGG cases that are falsely identified as HGG. We summarize the performance of each algorithm using a 2×2 confusion matrix that illustrates TP, TN, FP, and FN values, where each row of the confusion matrix represents an actual class, and each column corresponds to the predictions made by the models.

Sensitivity, also called true positive rate (TPR) or recall, is the ratio of true positives to the sum of all positive assessments. In our study, sensitivity measure gives the proportion of correctly predicted LGGs. The mathematical definition of sensitivity measure is given by:

$$Sensitivity = TPR = TP/(TP + FN) \tag{8}$$

Specificity, also known as true negative rate (TNR), gives the ratio of true negatives to the total number of negative assessments. This measure gives the proportion of correctly predicted HGG cases and is defined as:

$$Specificity = TNR = TN/(TN + FP) \tag{9}$$

False positive rate (FPR) gives the proportion of false positives out of all negative assessments. This provides the proportion of incorrectly labeled HGG cases and is given by:

$$FPR = FP/(FP + TN) \tag{10}$$

Accuracy is measured by the ratio of total number of correct assessments to the number of all assessments. This measure provides overall classification accuracy which corresponds to the proportion of correctly identified LGG and HGG cases and is formulated by:

$$Accuracy = (TP + TN)/(TP + TN + FP + FN) \tag{11}$$

Precision, also known as positive predictive value (PPV), is the proportion of true positives, i.e., ratio of correctly identified LGG cases, out of all the cases that the model classified as positive and is defined by:

$$Precision = TP/(TP + FP) \tag{12}$$

F1 score, also known as F measure, gives another insight into the classification accuracy by considering both precision and recall values and can be defined as weighted average of the precision and recall, is given by:

$$F1\ score = 2 * \frac{Precision * Recall}{Precision + Recall} = \frac{TP}{TP + (FP + FN)/2} \tag{13}$$

We also report the proportion of false positives against the true positives for each model by plotting the receiver operating characteristics (ROC) curves [54] of the cross validation runs. ROC curves help us roughly illustrate the diagnostic ability of each algorithm for each cross-validation run. So, we use ROC curves and the area under the curve (AUC) estimates as evaluation metrics as well.

4. Results

In this study, we focused on the classification of gliomas using deep CNN algorithms. A total of 104 pathology-proved cases (50 LGG and 54 HGG) were retrospectively acquired through T2W/ FLAIR MRI sequences and used for analysis. A group of stochastic, pipeline-based data augmentation techniques, such as flipping, rotating, translating, zooming, shifting and elastic transforms, were used to expand the training dataset to 20-fold larger in size. We applied a 5-fold cross validation procedure to our dataset. In the training process, at each fold, a random validation set, including 20% proportion from the training set, was used to tune the parameters of the models. A

separate test-set (20%) was used to evaluate the performance of the algorithms. The evaluation metrics were computed for each model by taking the average of the performance values measured at each cross validation run. The results show an overall high performance of both our custom model and the pretrained deep CNNs. The models correctly classified gliomas into LGG and HGG with an average AUC value of over 0.97. A comparison of classification accuracy for each model was listed in Table 5. According to the results, the custom-design deep CNN architecture achieved competitive or even better test performance measures than the pretrained CNN models explored in this study. The average sensitivity, specificity, precision, F1 score, accuracy, and AUC of the custom model amount to 0.980, 0.963, 0.961, 0.970, 0.971 and 0.989, respectively. GoogLeNet showed better performance than the other pretrained models and yielded a sensitivity, specificity, precision, F1 score, accuracy, and AUC values of 0.980, 0.889, 0.891, 0.933, 0.933 and 0.987, respectively. AlexNet demonstrated a sensitivity, specificity, precision, F1 score, accuracy, and AUC of 0.940, 0.907, 0.904, 0.922, 0.923 and 0.970, respectively. As for SqueezeNet, its performance was found to be similar to AlexNet in terms of the AUC value. The sensitivity, specificity, precision, F1 score, accuracy, and AUC values of SqueezeNet were 0.920, 0.870, 0.868, 0.893, 0.894, and 0.975, respectively. Figure 7 presents the resulting ROC curves with AUC values of the models.

Table 5. Performance of the models in classifying low and high-grade gliomas in terms of validation loss, sensitivity, specificity, precision, F1 score, test accuracy, and area under the receiver operating characteristics curves values.

Models	Loss	Sensitivity	Specificity	Precision	F1 score	Accuracy	AUC
Custom	0.300	0.980	0.963	0.961	0.970	0.971	0.989
AlexNet	0.364	0.940	0.907	0.904	0.922	0.923	0.970
GoogLeNet	0.328	0.980	0.889	0.891	0.933	0.933	0.987
SqueezeNet	0.345	0.920	0.870	0.868	0.893	0.894	0.975

The proposed model correctly predicted 49 out of 50 LGGs with two false positives and 52 out of 54 HGGs with one false negative. The numbers of correctly classified LGG cases for GoogLeNet, AlexNet and SqueezeNet were 49, 47 and 46, respectively. As for the HGGs, AlexNet, GoogLeNet and SqueezeNet correctly predicted 49, 48 and 47 cases, respectively. Cumulative confusion matrices of the classification results are given in Figure 8.

There were 16 cases (G1-G16) misclassified by at least one model as shown in Figure 9. Of those cases, G15 was not accurately identified by any of the four models. G1 and G12 were only correctly classified by one model. The findings suggest that some cases, especially associated with heterogeneous lesions containing cystic morphological formations, as found in G1, G12 and G15, are inclined to impair the classification performance of the models. Figure 9 shows the ground truth test cases where at least one of the models failed to correctly classify lesions.

The results showed that the models can effectively learn the feature representations and activations for HGGs and LGGs. The feature visualization on the classification layers of the models shows that the algorithms were able to create complex textures and patterns for each class. The visualization of LGG and HGG features learned by the models are presented in Figure 10–11. Additionally, the visualization of a set of learned activations on LGG and HGG inputs are illustrated for each model in Figure 12.

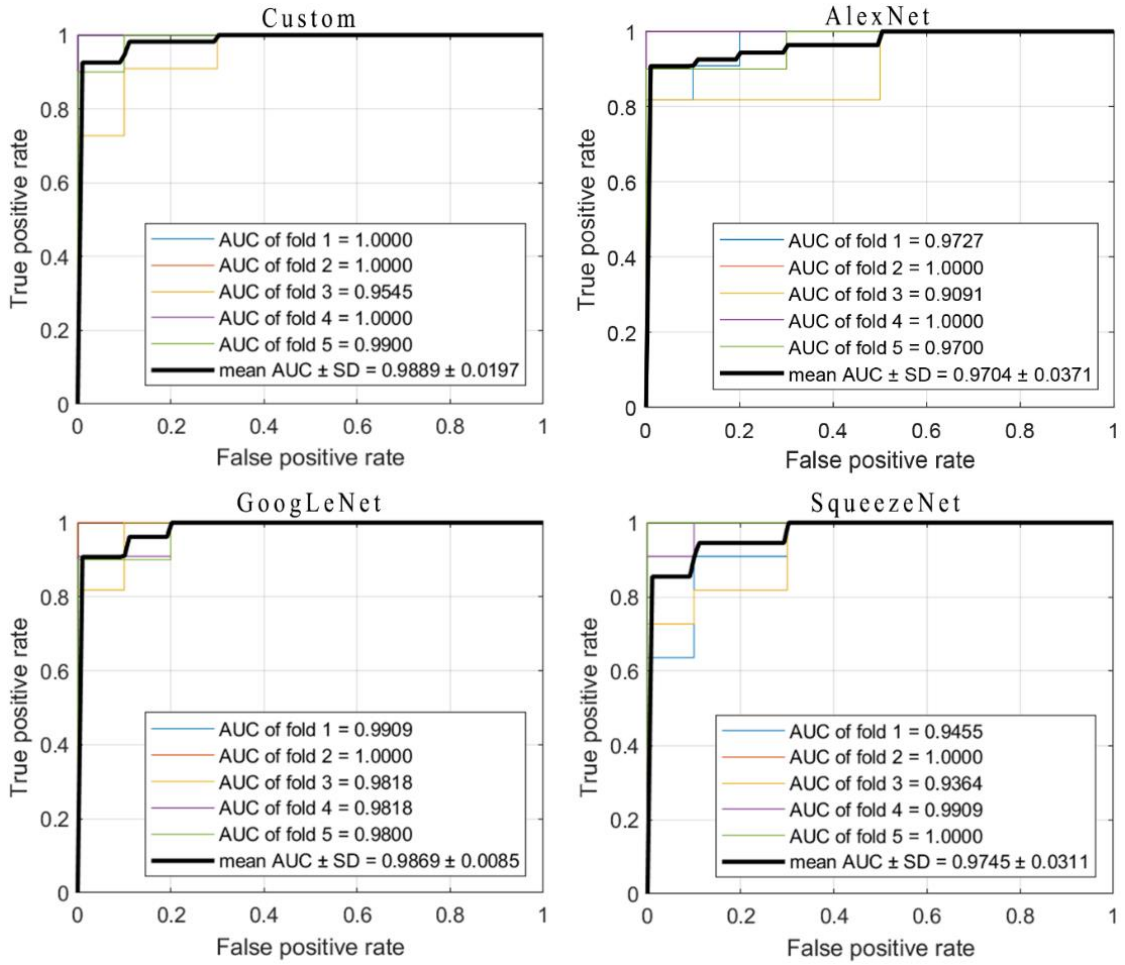

Figure 7. The classification performance of the custom model, AlexNet, GoogLeNet and SqueezeNet in terms of the area under the receiver operating characteristics curves with the pooled (mean) performance over 5-fold cross-validation runs. This figure indicates how the classification performance is affected by each training process, which is shown by colored lines.

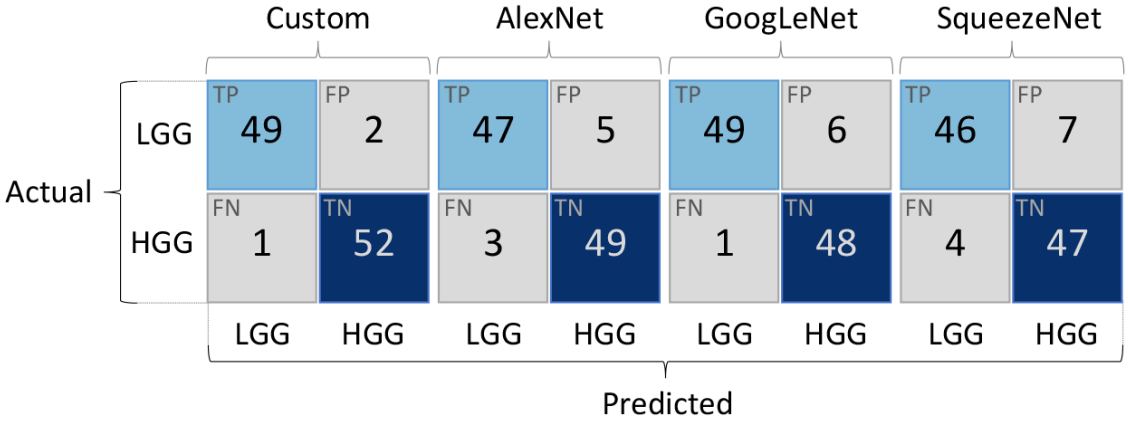

Figure 8. Cumulative confusion matrices of the classification results of a 5-fold cross-validation using the custom model, AlexNet, GoogLeNet and SqueezeNet.

Figure 9. Ground truth test cases (G1-16) where at least one of the models misclassified lesions. Predicted classes are shown in percentages for each model. Models which correctly classified lesions are in green color. Models which failed to correctly classify lesions are shown in red color. Cystic formations in G1, G12 and G15 are roughly outlined by the dashed yellow rectangular boxes.

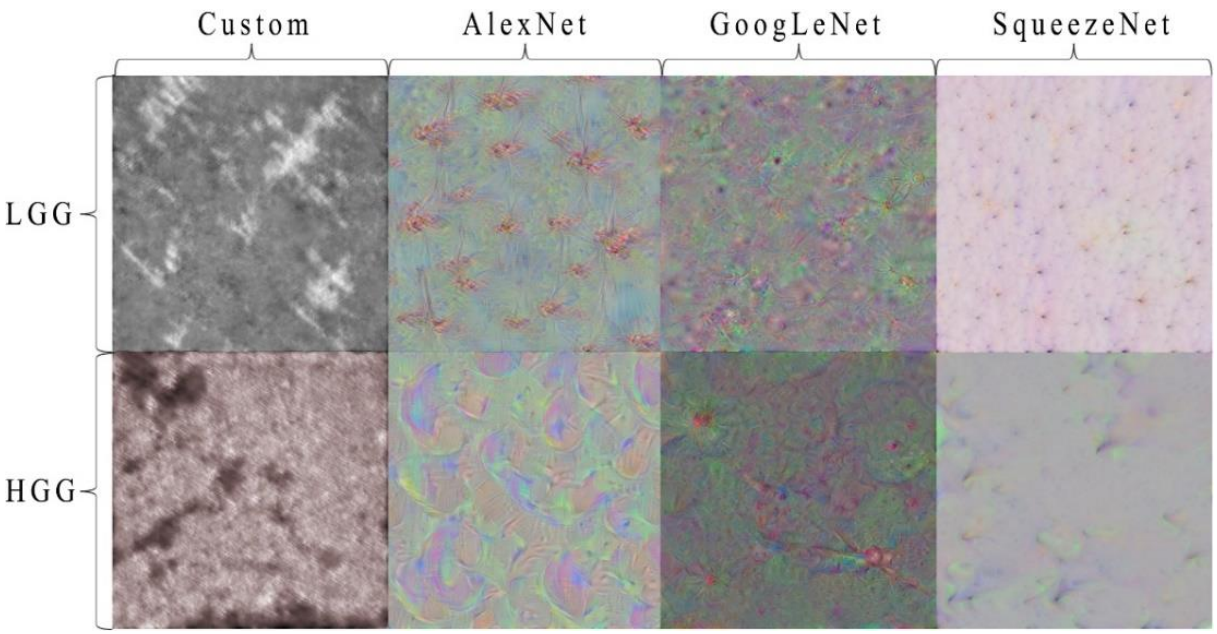

Figure 10. Visualization of LGG and HGG features learned by the models.

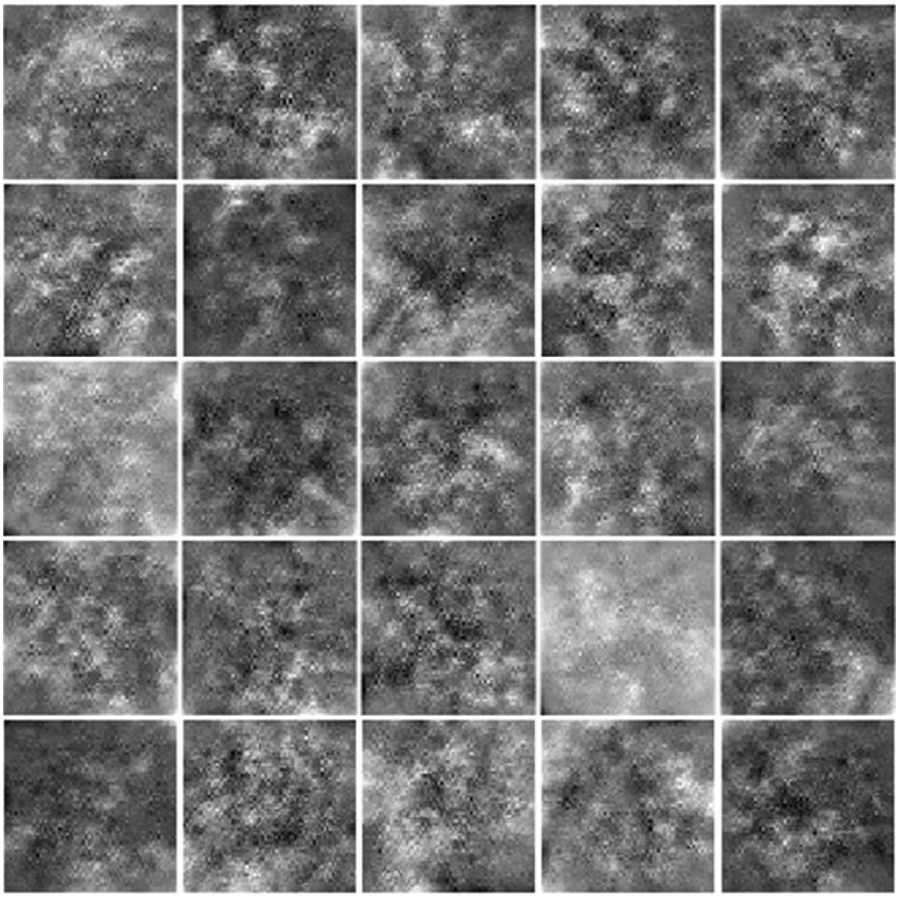

Figure 11. A set of features learned by the last ReLU layer of the custom model.

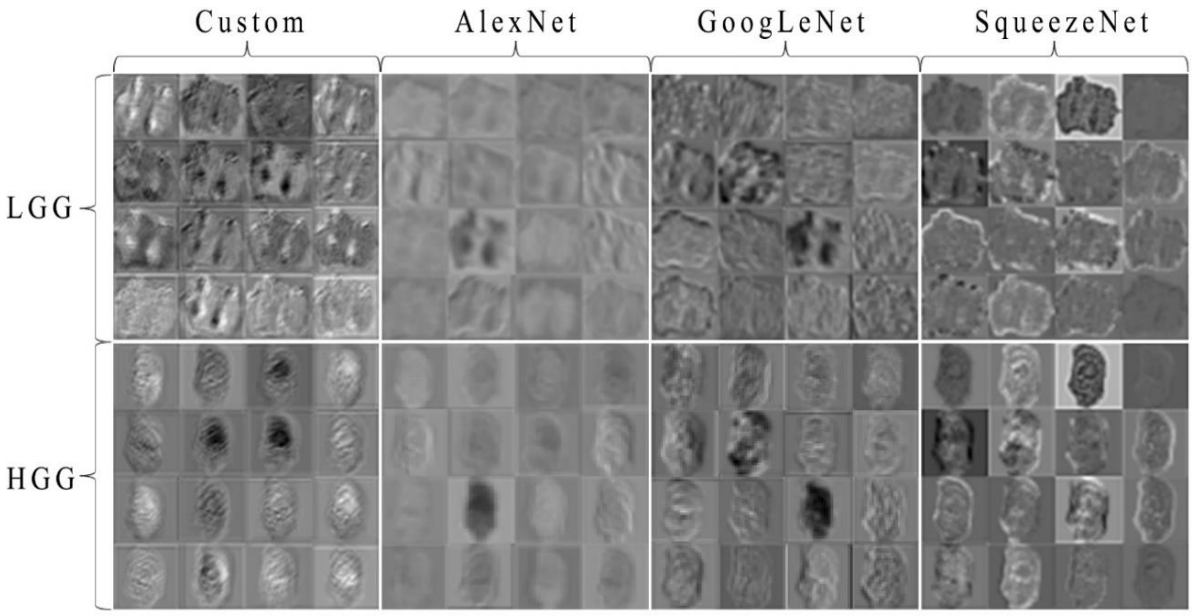

Figure 12. Visualization of the activations for each model on low-grade glioma (G4 in Figure 9) and high-grade glioma (G1 in Figure 9) inputs. This figure illustrates the first 16 activations learned by the convolutional layers "conv 3", "conv 2", "inception 3a 3x3", and "fire 4 squeeze 1x1" of the custom model, AlexNet, GoogLeNet and SqueezeNet, respectively.

The total running time of the program for training the models was 41.7 hours. Table 6 presents the computational complexity of the trained models in terms of memory size, training time, number of epochs and iterations. All the reported values were averaged over 5-fold cross validation runs. The size of the custom model, AlexNet, GoogLeNet and SqueezeNet are 3.00MB, 217.02MB, 23.12MB and 2.91MB, respectively. The custom model, AlexNet, GoogLeNet and SqueezeNet were trained with 1556.40, 658.40, 484.00, 578.20 iterations, within 39.20, 16.60, 12.80 and 15.00 epochs, respectively. The average training time spent (in minutes) for the custom model, AlexNet, GoogLeNet and SqueezeNet were 19.58, 123.32, 237.74, and 120.30, respectively.

Table 6. Computational complexity of training the custom model, AlexNet, GoogLeNet and SqueezeNet over the dataset.

Model	Memory size (MB)	Elapsed time (minutes)	Epoch	Iteration
Custom	3.00	19.58 ± 0.92	39.20 ± 1.10	1556.40 ± 60.12
AlexNet	217.02	123.32 ± 14.89	16.60 ± 2.19	658.40 ± 96.76
GoogLeNet	23.12	237.74 ± 64.48	12.80 ± 3.27	484.00 ± 130.36
SqueezeNet	2.91	120.30 ± 33.02	15.00 ± 3.39	578.20 ± 146.51

*Note. For the pretrained models, memory size represents the size after fine-tuning. Elapsed time and iteration values were averaged over 5-fold cross validation runs.

5. Conclusions

In this study, a new convolutional neural network (CNN) based deep learning model was

proposed and its performance compared with the implementation of pretrained AlexNet, GoogLeNet and SqueezeNet through transfer learning for the glioma grade analysis. A group of pathology-proven 104 clinical cases with glioma were used as a dataset. Data augmentation techniques were used to expand the training data. Five-fold cross-validation was applied to evaluate the classification ability of each model. Results showed that a robust CNN model can be developed and trained on a clinical MRI dataset, even with a small size. The proposed deep CNN model achieved comparable or even better performance than the pretrained models in terms of sensitivity, specificity, F1 score, accuracy, and AUC values. The training of the proposed model took place ~6 times faster than SqueezeNet, which was trained in the shortest time compared to the other pretrained networks used in the study. The experimental results revealed that the proposed model was lightweight, yet advantageous in terms of both accuracy and computational complexity.

Although this study provided additional insight into CNN-based applications for glioma analysis through clinical MRI images, several limitations must be noted, including its retrospective design and a small dataset. More studies are needed to further assess our findings. Future studies may use different MRI protocols with multimodal designs and explore protocol-specific factors influential on the classification. New models can be built to accept multi-scanner outputs and further evaluation of the methods on publicly available datasets can be investigated. The findings can be used to assist the development of clinical decision support systems.

Acknowledgments

All authors declare that this study received no financial support. This retrospective study was granted the institutional research ethics committee approval (15386878-044/2020) and adhered to the tenets of the Declaration of Helsinki. This paper is a part of doctoral research of the first author at the Department of Computer Engineering at Kırıkkale University.

Conflict of interest

All authors declare no conflicts of interest in this paper.

Data availability

The materials/data used to support the findings of this study are available from the corresponding author upon reasonable request.

References

1. R. Chen, M. Smith-Cohn, A. L. Cohen, H. Colman, Glioma subclassifications and their clinical significance, *Neurotherapeutics*, **14** (2017), 284–297.
2. Y.-C. Liu, Y. Wang, Role of yes-associated protein 1 in gliomas: Pathologic and therapeutic aspects, *Tumor Biol.*, **36** (2015), 2223–2227.
3. D. Persaud-Sharma, J. Burns, J. Trangle, S. Moulik, Disparities in brain cancer in the united states: A literature review of gliomas, *Med. Sci. Basel Switz.*, **5** (2017), 16.
4. D. N. Louis, A. Perry, G. Reifenberger, A. von Deimling, D. Figarella-Branger, W. K. Cavenee, et al., The 2016 World Health Organization classification of tumors of the central nervous system: A

summary, *Acta Neuropathol.*, **131** (2016), 803–820.
5. C. Walker, A. Baborie, D. Crooks, S. Wilkins, M. D. Jenkinson, Biology, genetics and imaging of glial cell tumours, *Br. J. Radiol.*, **84** (2011), S90–S106.
6. F. Dhermain, Radiotherapy of high-grade gliomas: current standards and new concepts, innovations in imaging and radiotherapy, and new therapeutic approaches, *Chin. J. Cancer*, **33** (2014), 16–24.
7. E. M. Sizoo, L. Braam, T. J. Postma, H. R. W. Pasman, J. J. Heimans, M. Klein, et al., Symptoms and problems in the end-of-life phase of high-grade glioma patients, *Neuro-Oncol.*, **12** (2010), 1162–1166.
8. R. Stupp, W. P. Mason, M. J. van den Bent, M. Weller, B. Fisher, M. J. B. Taphoorn, et al., Radiotherapy plus concomitant and adjuvant temozolomide for glioblastoma, *N. Engl. J. Med.*, **352** (2005), 987–996.
9. Q. T. Ostrom, L. Bauchet, F. G. Davis, I. Deltour, J. L. Fisher, C. E. Langer, et al., The epidemiology of glioma in adults: A "state of the science" review, *Neuro-Oncol.*, **16** (2014), 896–913.
10. E. B. Claus, K. M. Walsh, J. K. Wiencke, A. M. Molinaro, J. L. Wiemels, J. M. Schildkraut, et al., Survival and low-grade glioma: The emergence of genetic information, *Neurosurg. Focus*, **38** (2015), E6.
11. K. S. Patel, B. S. Carter, C. C. Chen, Role of biopsies in the management of intracranial gliomas, *Prog. Neurol. Surg.*, **30** (2018), 232–243.
12. R. J. Jackson, G. N. Fuller, D. Abi-Said, F. F. Lang, Z. L. Gokaslan, W. M. Shi, et al., Limitations of stereotactic biopsy in the initial management of gliomas, *Neuro-Oncol.*, **3** (2001), 193–200.
13. M. Preusser, K. Aldape, E. Gerstner, W. Pope, M. Viapiano, Highlights from the literature, *Neuro-Oncol.*, **19** (2017), 1154–1157.
14. J. Zhang, H. Liu, H. Tong, S. Wang, Y. Yang, G. Liu, et al., Clinical applications of contrast-enhanced perfusion MRI techniques in gliomas: Recent advances and current challenges, *Contrast Media Mol. Imaging*, **2017** (2017), 1–27.
15. E. Moser, A. Stadlbauer, C. Windischberger, H. H. Quick, M. E. Ladd, Magnetic resonance imaging methodology, *Eur. J. Nucl. Med. Mol. Imaging*, **36** (2009), 30–41.
16. A. Patra, A. Janu, A. Sahu, MR Imaging in neurocritical care, *Indian J. Crit. Care Med. Peer-Rev. Off. Publ. Indian Soc. Crit. Care Med.*, **23** (2019), S104–S114.
17. S. Waite, J. Scott, B. Gale, T. Fuchs, S. Kolla, D. Reede, Interpretive error in radiology, *Am. J. Roentgenol.*, **208** (2017), 739–749.
18. F. Caranci, E. Tedeschi, G. Leone, A. Reginelli, G. Gatta, A. Pinto, et al., Errors in neuroradiology, *Radiol. Med.*, **120** (2015), 795–801.
19. Y. Kang, S. H. Choi, Y.-J. Kim, K. G. Kim, C.-H. Sohn, J.-H. Kim, et al., Gliomas: Histogram analysis of apparent diffusion coefficient maps with standard- or high-b-value diffusion-weighted MR imaging--correlation with tumor grade, *Radiology*, **261** (2011), 882–890.
20. G. Ranjith, R. Parvathy, V. Vikas, K. Chandrasekharan, S. Nair, Machine learning methods for the classification of gliomas: Initial results using features extracted from MR spectroscopy, *Neuroradiol. J.*, **28** (2015), 106–111.
21. F. P. Polly, S. K. Shil, M. A. Hossain, A. Ayman, Y. M. Jang, *Detection and classification of HGG and LGG brain tumor using machine learning*, Proceedings of the 32nd International Conference on Information Networking, Thailand, 2018.

22. Q. Tian, L.-F. Yan, X. Zhang, X. Zhang, Y.-C. Hu, Y. Han, et al., Radiomics strategy for glioma grading using texture features from multiparametric MRI: Radiomics approach for glioma grading, *J. Magn. Reson. Imaging*, **48** (2018), 1518–1528.
23. X. Bi, J. G. Liu, Y. S. Cao, *Classification of low-grade and high-grade glioma using multiparametric radiomics model*, Proceedings of the 3rd IEEE Information Technology, Networking, Electronic and Automation Control Conference, China, 2019.
24. G. Cui, J. Jeong, B. Press, Y. Lei, H.-K. Shu, T. Liu, et al., Machine-learning-based classification of lower-grade gliomas and high-grade gliomas using radiomic features in multi-parametric MRI, preprint, arXiv:1911.10145.
25. A. S. Lundervold, A. Lundervold, An overview of deep learning in medical imaging focusing on MRI, *Z. Phys.*, **29** (2019), 102–127.
26. K. Fukushima, A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position, *Biol. Cybern.*, **36** (1980), 193–202.
27. J. Gao, Q. Jiang, B. Zhou, D. Chen, Convolutional neural networks for computer-aided detection or diagnosis in medical image analysis: an overview, *Math. Biosci. Eng.*, **16** (2019), 6536–6561.
28. J. Yosinski, J. Clune, Y. Bengio, H. Lipson, How transferable are features in deep neural networks?, preprint, arXiv:1411.1792.
29. E. I. Zacharaki, S. Wang, S. Chawla, D. Soo Yoo, R. Wolf, E. R. Melhem, et al., Classification of brain tumor type and grade using MRI texture and shape in a machine learning scheme, *Magn. Reson. Med.*, **62** (2009), 1609–1618.
30. A. Ditmer, B. Zhang, T. Shujaat, A. Pavlina, N. Luibrand, M. Gaskill-Shipley, et al., Diagnostic accuracy of MRI texture analysis for grading gliomas, *J. Neurooncol.*, **140** (2018), 583–589.
31. S. Banerjee, S. Mitra, F. Masulli, S. Rovetta, Deep radiomics for brain tumor detection and classification from multi-sequence MRI, preprint, arXiv:1903.09240.
32. Y. Zhuge, H. Ning, P. Mathen, J. Y. Cheng, A. V. Krauze, K. Camphausen, et al., Automated glioma grading on conventional MRI images using deep convolutional neural networks, *Med. Phys.*, **47** (2020), 3044-3053.
33. E. Lotan, R. Jain, N. Razavian, G. M. Fatterpekar, Y. W. Lui, State of the art: Machine learning applications in glioma imaging, *Am. J. Roentgenol.*, **212** (2019), 26–37.
34. P. Korfiatis, B. Erickson, Deep learning can see the unseeable: Predicting molecular markers from MRI of brain gliomas, *Clin. Radiol.*, **74** (2019), 367–373.
35. R. Takahashi, T. Matsubara, K. Uehara, Data augmentation using random image cropping and patching for deep CNNs, *IEEE Trans. Circuits Syst. Video Technol.*, **30** (2020), 2917–2931.
36. J. Ding, X. Li, X. Kang, V. N. Gudivada, A case study of the augmentation and evaluation of training data for deep learning, *J. Data Inf. Qual.*, **11** (2019), 1–22.
37. O. Fink, Q. Wang, M. Svensén, P. Dersin, W.-J. Lee, M. Ducoffe, Potential, challenges and future directions for deep learning in prognostics and health management applications, *Eng. Appl. Artif. Intell.*, **92** (2020), 103678.
38. M. D. Bloice, C. Stocker, A. Holzinger, Augmentor: An image augmentation library for machine learning, preprint, arXiv:1708.04680.
39. G. Liu, K. J. Shih, T.-C. Wang, F. A. Reda, K. Sapra, Z. Yu, et al., Partial convolution based padding, preprint, arXiv:1811.11718.
40. S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, preprint, arXiv:1502.03167.

41. J. L. Ba, J. R. Kiros, G. E. Hinton, Layer normalization, preprint, arXiv:1607.06450.
42. K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification, preprint, arXiv:1502.01852.
43. B. Xu, N. Wang, T. Chen, M. Li, Empirical evaluation of rectified activations in convolutional network, preprint, arXiv:1505.00853.
44. C. Banerjee, T. Mukherjee, E. Pasiliao, *An empirical study on generalizations of the ReLU activation function*, Proceedings of the 20th ACM Conference on Economics and Computation, USA, 2019.
45. M. Ranzato, F. J. Huang, Y.-L. Boureau, Y. LeCun, *Unsupervised learning of invariant feature hierarchies with applications to object recognition*, Proceedings of the 25th IEEE Conference on Computer Vision and Pattern Recognition, USA, 2007.
46. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: A simple way to prevent neural networks from overfitting, *J. Mach. Learn. Res.*, **15** (2014), 1929–1958.
47. L. Bottou, Stochastic gradient descent tricks, in *Neural Networks: Tricks of the Trade*, (eds. G. Montavon, G. B. Orr, and K.-R. Müller), Springer Berlin Heidelberg, (2012), 421–436.
48. J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, *ImageNet: A large-scale hierarchical image database*, Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition, USA, 2009.
49. A. Krizhevsky, I. Sutskever, G. E. Hinton, ImageNet classification with deep convolutional neural networks, *Commun. ACM*, **60** (2017), 84–90.
50. C. Szegedy, Wei Liu, Yangqing Jia, P. Sermanet, S. Reed, D. Anguelov, et al., *Going deeper with convolutions*, Proceedings of the 28th IEEE Conference on Computer Vision and Pattern Recognition, USA, 2015.
51. F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, K. Keutzer, SqueezeNet: AlexNet-level accuracy with 50× fewer parameters and <0.5mb model size, preprint, arXiv:1602.07360.
52. G. Raskutti, M. J. Wainwright, B. Yu, Early stopping and non-parametric regression: An optimal data-dependent stopping rule, *J. Mach. Learn. Res.*, **15** (2014), 335–266.
53. L. Prechelt, Early stopping - but when?, in *Neural Networks: Tricks of the Trade*, (eds. G. B. Orr and K.-R. Müller), Springer Berlin Heidelberg, (1998), 55–69.
54. J. A. Hanley, B. J. McNeil, The meaning and use of the area under a receiver operating characteristic (ROC) curve., *Radiology*, **143** (1982), 29–36.

AIMS Press