International Journal of Innovative Engineering Applications

Journal homepage: https://dergipark.org.tr/ijiea

CLASSIFICATION OF BRAIN TUMORS USING ARTIFICIAL INTELLIGENCE

Sedat Bayaral*[*1], Evrim Gül[2], Derya Avcı[3]

[1]*Firat University, Instutie of Science, Sofware Engineering, Elazig, Turkey*
[2]*Firat University, Faculty of Medicine, Department of Emergency Medicine, Elazig, Turkey*
[3]*Firat University, Vocational School of Technical Sciences, Department of Computer Technologies, Elazig, Turkey*

Abstract

Original scientific paper

Brain MRI is a medical image obtained by MRI, which stands for "Magnetic Resonance Imaging". Brain MRI uses magnetic fields and radio waves to create detailed images of the brain and surrounding tissues. Today, deep learning algorithms are used to detect brain tumors or classify different brain regions. In this study, feature extraction has been performed with current deep learning models using a dataset consisting of 7023 open access images obtained from patients from various parts of the world, and the results were evaluated by training Support Vector Machine (SVM) and XGBoost models with the extracted features. In this study, 4 deep learning models, VGG16, VGG19, ResNet50 and MobileNetV2, have been used for feature extraction. In order to achieve higher performance, transfer learning method is used in this study, which allows the weights of models that are pre-trained with large data sets to be used in other models. The weights of the models trained with ImageNet were included in the study to improve performance and save time. Although the original layer structures of the models are fixed, the GlobalAveragePooling2D layer has been added to the CNN models to improve performance and generalize the features extracted from deep learning models. Brain MRI images divided into 4 classes as glioma tumor, meningioma tumor, pituitary tumor and no tumor. Auxiliary functions have been used to obtain optimum values for the parameters used for training the models. Accuracy, F1-score, precision and sensitivity metrics used to evaluate the training results. When the results are evaluated, the best performance with an F1-score of 97.87% is obtained by classifying the features extracted from the ResNet50 CNN model with Support Vector Machine (SVM).

Keywords: *Artificial intelligence, machine learning, brain tumor MRI, classification.*

1 Introduction

Cells that grow uncontrollably in the skull region are called brain tumors. Early diagnosis and diagnosis of brain tumors is very important for the treatment of the disease. In recent years, various medical imaging systems have been developed to detect brain tumors. One of them, Brain MRI, is a medical imaging method also known as "Magnetic Resonance Imaging". Brain MRI uses magnetic fields and radio waves to create detailed images of the brain and surrounding tissues. These images provide important information for diagnosing and monitoring brain diseases and planning treatment. Brain tumors are complex and a professional neurosurgeon is required to identify the tumor and its type from an MRI image. Diagnosing tumors from MRI with the help of artificial intelligence systems provides time and convenience to experts.

Today, deep learning algorithms are used to detect brain tumors or classify different brain regions. Deep learning is a machine learning method that can predict subsequent inputs by feeding based on available data. The quantity and quality of data affect the performance of deep learning models. Among with other medical imaging techniques, MRI allows for clearer images. Therefore, thanks to MRI which can display tumors developing in the brain in detail, deep learning models have given more precise results, and studies in this field have attracted the attention of researchers, and many new and productive studies have been carried out in this field.

2 Related Works

The use of artificial intelligence technologies in the diagnosis of medical diseases has started to increase in recent years. Developments in medical imaging systems and the increasingly high quality and detailed images obtained from these imaging systems have attracted the attention of researchers. MRI, one of the medical imaging systems, is used to detect brain tumors. Artificial intelligence models, which can successfully segment and classify images by minimizing the error, facilitate the work of experts in the early diagnosis of the disease. Convolutional Neural Networks (CNN), one of the most popular deep learning models in image classification in recent years, are frequently used in the classification of medical images. For this reason, various studies have been conducted on tumor detection from brain MRI using artificial intelligence technologies.

*Corresponding author.
E-mail address: sedatbayaral97@hotmail.com (S. Bayaral)
Received 08 October 2024; Received in revised form 02 February 2025; Accepted 06 March 2025
 Doi: https://doi.org/10.46460/ijiea.1563426

Karamehić et al. [1] utilized a dataset of 4 classes and preprocessed the images in the dataset with the Python PIL library. After the classification process with VGG16, an accuracy of 96.9% was achieved.

Remzan et al. [2] conducted a classification study using a dataset consisting of 5712 images. In the study, 7 CNN models were incorporated through transfer learning, and feature extraction was performed using these models. The extracted features were evaluated with various machine learning classifiers, and a feature ensemble was created using the top 3 features with the best performance. Subsequently, the best-performing machine learning classifier, MLP (Multi-Layer Perceptron), was employed for brain MRI classification. The study revealed that the best features were obtained from the ResNet50, VGG19, and EfficientNetV2B1 CNN models. Finally, an accuracy of 96.67% was achieved in classification using MLP.

Pal et al. [3] conducted a classification study using a 4-class dataset. The VGG16 CNN model was employed in the study, and when the results were evaluated using the AUC (Area Under ROC Curve) metric, a success rate of 92% was recorded. The model achieved a success rate of 90% for the meningioma class, 91% for the pituitary class, 93% for the glioma class, and 89% for the non-tumor class.

Bohra et al. [4] utilized 3 CNN models for feature extraction and classified the extracted features using various machine learning algorithms. It was noted that the best results were obtained from the features extracted with the aid of the VGG16 CNN model, achieving an accuracy of 97.7%.

Hong [5] classified Brain MRI using a basic CNN model and VGG16. After 300 epochs, the basic CNN model achieved an accuracy of 67%, while VGG16 achieved an accuracy of 80%.

Latif et al. [6] performed feature extraction using the 3D Discrete Wavelet Transform and conducted classification with a Random Forest classifier, also comparing the results with various other classification algorithms. The study results indicated that high-grade glioma was detected with an average accuracy of 89.75%, while low-grade glioma was detected with an average accuracy of 86.7%.

Kibriya et al. [7] performed feature extraction using deep learning models and conducted classification with Support Vector Machine (SVM) from the machine learning models. A total of 15320 MRI images were utilized in the study, achieving an accuracy of 98% with the CNN-SVM based model.

Tandel et al. [8], compared transfer learning-based CNN models with traditional machine learning algorithms, including Decision Trees, Linear Discriminant Analysis, Naive Bayes, SVM, K-Nearest Neighbors (KNN), and ensemble methods. The comparison of the AlexNet model with these 6 machine learning algorithms revealed that deep learning was 12% more successful than machine learning.

Kumar et al. [9] utilized a dataset of 3064 brain MRI images. ResNet50 was employed for classification, resulting in mean accuracy of 97.08% with image augmentation and 97.48% without image augmentation.

Gürkahraman et al. [10] performed brain tumor detection from T1-weighted MRI. In the study, DenseNet121 with transfer learning, one of the ANN models, was utilized for feature extraction, and the obtained features were classified using SVM, KNN, and Bayesian algorithms. The results indicated that the Pearson correlation coefficient (R) of the CNN architecture increased from 88% to 96.7%, and the accuracy improved from 94.6% to 98.6% after image augmentation. The accuracy values of the SVM, KNN, and Bayesian classifiers were recorded as 99.8%, 99.1%, and 89.3%, respectively, while the Pearson correlation coefficient values were 99.5%, 98.9%, and 79%, respectively.

Taşçı [11] utilized a dataset consisting of 3443 images with 4 classes. Among the 19 pre-trained models, the features extracted from the DenseNet201 model yielded the highest accuracy. In the proposed method, 1920 features from the 'avg_pool' layer and 1000 features from the 'fc1000' layer of DenseNet201 were combined. An accuracy of 95.00% was achieved without feature selection, while an accuracy of 95.76% was obtained with mRMR feature selection.

Uysal et al. [12] utilized AlexNet, VGG16, and MobileNetV3 models for classifying a 4-class dataset. The accuracy rates of the images classified using AlexNet, VGG16, and MobileNetV3 were 94.47%, 96.875%, and 95.673%, respectively, while the F1-scores were 94%, 97%, and 96%, respectively.

Demir et al. [13] utilized a pre-trained Convolutional Neural Network model called MobileNetV2 for feature extraction. Subsequently, the ReliefF algorithm was employed for feature selection. The features extracted by MobileNetV2 and those selected by the ReliefF algorithm were fed into the classifiers separately, and the system performance was evaluated. The experimental results indicated that the highest performance was achieved with the combination of MobileNetV2 for feature extraction, ReliefF for feature selection, and the KNN classifier.

Aslan [14] utilized the KNN algorithm in combination with the MobileNetV2 deep learning model for the detection of brain tumors using MRI. In the study, the values from the fully connected layers of the pre-trained MobileNetV2 model were used as features. The KNN classification algorithm was employed to enhance the classification performance of the extracted features. In the experimental studies, an accuracy score of 96.44% was achieved with the proposed method using the KNN classifier for the detection of brain tumors.

Paul et al. [15] classified MRI using models based on Fully Connected and Convolutional Neural Networks architectures. The prediction results in one of three classes: meningioma, glioma, or pituitary. A total of 989 axial images from 191 patients were used (to avoid confusing the neural networks with three different planes containing the same diagnosis). The 512x512 images were augmented, and the models were tested. As a result of the 5-fold cross-validation evaluation, the highest performance rate was found to be a mean accuracy of 91.43%.

Srinivas [16] proposed a model combining CNN and KNN. Feature extraction was conducted using CNN models, while classification was carried out using KNN models. The BraTS 2015 and BraTS 2017 datasets were

employed in the study. As a result, an accuracy of 96.25% was achieved.

Jayade et al. [17] extracted features using the GLCM (Gray Level Co-occurrence Matrix) method and classified benign and malignant tumors with SVM and KNN classifiers. The study utilizing only the SVM classifier achieved an accuracy of 91.21%, while the study using only the KNN classifier achieved an accuracy of 79.23%. However, an accuracy of 94.13% was attained in the study conducted with the new model created by combining both classifiers.

Ayadi et al. [18] investigated the detection of three types of brain tumors. The SVM classifier was employed, achieving an accuracy of 90.27%. The results were obtained through k-fold cross-validation.

Amin et al. [19] conducted classification using T1C, T1, Flair, and T2 weighted MRI. Information with varying textures and structures was extracted from these images of different weights. In order to fuse them, Daubechies wavelet kernels and DWT were employed for more informative tumor region detection. Following this fusion process, noise removal was applied using PDDF. The BRATS 2012, BRATS 2013, BRATS 2015, BRATS 2013 leaderboard, and BRATS 2018 datasets were utilized to evaluate the proposed method. As a result of the study, the fusion of information obtained from different imaging techniques yielded more successful outcomes compared to single sequences.

Shahin et al. [20] proposed a new neural network architecture comprising 4 modules. In the first module, feature extraction is performed. In the next module, RSPA is utilized to enhance the network's features and emphasize significant regions. The next module, ASPP, is utilized for the collection and integration of features at different scales. And the last module is utilized for classification. In the study, 4 publicly available datasets containing 9581 images were utilized. Evaluation with the Cheng, Brats-small-2C, and Brats-small-4C datasets yielded accuracies of 99.78%, 99.33%, and 96.33%, respectively.

Kang et al. [21] employed transfer learning models for feature extraction and evaluated these features using 7 different machine learning algorithms. The top three feature sets that demonstrated optimal performance among the machine learning algorithms were selected and subsequently provided as input to various machine learning models for classification. The results indicated that Support Vector Machine (SVM) classifiers, particularly those utilizing radial basis function kernels, exhibited strong performance, especially in large datasets.

Guzmán et al. [22] conducted a classification study of brain MRI using 7 CNN models. The images comprised four classes: glioma, meningioma, pituitary, and tumor-free. A dataset of 7023 images was created by combining the Fighshare, SARTAJ, and Br35H datasets. Classification was performed using one traditional CNN model and six pre-trained CNN models. The models employed in the study included the traditional CNN implementation, ResNet50, InceptionV3, InceptionResNetV2, Xception, MobileNetV2, and EfficientNetB0. The best performance was achieved with InceptionV3, yielding an accuracy of 97.12%.

Tiwari et al. [23] conducted a classification study utilizing a dataset consisting of four classes: meningioma, glioma, pituitary, and tumor-free. In the study, a new CNN model was proposed, and an accuracy of 99% was achieved following the classification process with this model.

Aamir et al. [24] conducted a classification study utilizing brain MRI. Firstly, image quality enhancement was performed. In the second stage, feature extraction was carried out using two different pre-trained deep learning models. The PLS method was employed to convert these features into a hybrid feature vector form. In the third stage, top tumor locations were extracted through agglomerative clustering. Subsequently, these results were resized appropriately and fed into the main model for classification. The results indicated that the proposed model achieved a classification accuracy of 98.95% compared to existing models.

Saleh et al. [25] conducted a study to detect brain tumors and their types from MRI utilizing five pre-trained models. The models employed in the study included Xception, ResNet50, InceptionV3, VGG16, and MobileNet, with F1-score values obtained as 98.75%, 98.50%, 98.00%, 97.50%, and 97.25%, respectively. These accuracy rates are reported to have a positive impact on the early detection of tumors, potentially preventing paralysis and other disabilities.

Muhammad et al. [26] utilized VGG16 and ResNet50 models to classify brain MRI, evaluating the results based on metrics such as accuracy, sensitivity, specificity, and ROC curve analysis. The VGG16 model achieved 96% accuracy in classifying brain tumors, while the classification of malignant tumors yielded an accuracy of 94.30% when compared to the ResNet50 model. Conversely, the ResNet50 model demonstrated a classification accuracy of 93.10% for benign tumors in comparison with the VGG16 model.

Bahya et al. [27] employed machine learning algorithms to automate brain tumor detection in MRI scans, diagnosing normal and abnormal cases, including meningioma, glioma, and pituitary tumors. Feature extraction techniques such as FFT, K-means clustering, and Tamura texture analysis were utilized, along with dimensionality reduction methods. The study achieved an accuracy of 92.6% with Gradient Boosting (GB), 86% with Adaptive Boosting (ADA), and 82% with SVM for normal and abnormal classification. For specific tumor types, GB achieved 67.9%, SVM 65.3%, and ADA 59.6%. Due to its superior performance, GB was recommended as the most effective algorithm.

Dewan et al. [28] investigated brain tumor detection using MRI scans, focusing on glioma, meningioma, and pituitary tumors. Feature extraction was performed using the Gray Level Co-occurrence Matrix (GLCM) method, and the extracted features were classified using machine learning algorithms, including KNN, SVM, Decision Tree, Naïve Bayes, Logistic Regression, and Random Forest. Among these, Random Forest achieved the highest accuracy of 91.04%. The proposed approach effectively distinguishes between different tumor types and normal cases, aiding in early diagnosis and treatment planning.

Akter et al. [29] proposed a deep learning-based approach for automatic brain tumor classification and

segmentation using MRI scans. A CNN-based model was developed for classifying brain images into four categories, while a U-Net-based model was employed for tumor segmentation. The study utilized six benchmark datasets to evaluate the impact of segmentation on classification performance. Two classification methods were assessed based on accuracy, recall, precision, and AUC. The proposed model outperformed existing pre-trained models, achieving a highest accuracy of 98.7% on a merged dataset and 98.8% with segmentation. The results suggest that this framework could be effectively used in clinical settings for automatic brain tumor detection and segmentation.

N. et al. [30] developed a multiclass classification framework for brain tumor detection using a standard CNN architecture and transfer learning. The study classified meningioma, glioma, pituitary tumors, and normal cases by utilizing pre-trained models such as VGG16, AlexNet, and ResNet50, which were fine-tuned on a balanced dataset. Various optimizers, including Adam, AdaDelta, and SGD, were evaluated to enhance performance. The highest accuracy of 99.83% was achieved using VGG16 with the AdaDelta optimizer, while ResNet50 had the lowest accuracy of 70%. The custom CNN model demonstrated the lowest loss of 0.04%. The findings highlight the potential of deep learning-based models in improving brain tumor diagnosis and treatment planning.

Filatov et al. [31] addressed the challenge of manual brain tumor diagnosis by utilizing pretrained convolutional neural networks (CNN) for automated classification. The study classified three tumor types along with non-tumor MRI images using models such as ResNet50, EfficientNetB1, EfficientNetB7, and EfficientNetV2B1. Among these, EfficientNet demonstrated strong performance due to its scalable nature, with EfficientNetB1 achieving the highest accuracy—87.67% for training and 89.55% for validation. The findings highlight the potential of deep learning in reducing diagnostic errors and improving brain tumor classification.

Islam et al. [32] investigated the effectiveness of deep transfer learning architectures in brain tumor diagnosis using MRI scans. The study applied four pretrained models—InceptionV3, VGG19, DenseNet121, and MobileNet—on a dataset compiled from three benchmark sources: Figshare, SARTAJ, and Br35H. The dataset consisted of four classes: pituitary tumor, meningioma, glioma, and no tumor, with image augmentation applied to balance class distributions. Experimental results demonstrated that MobileNet achieved the highest accuracy of 99.60%, outperforming other models. These findings emphasize the potential of transfer learning in enhancing brain tumor classification accuracy.

Ullah et al. [33] explored the classification of brain tumors using deep learning-based methods on MRI images. The study addressed the challenge of dataset imbalance, which can bias classifier performance, by generating synthetic images using a sparse autoencoder network. Two pretrained neural networks were fine-tuned with Bayesian optimization, and deep features were extracted from the global average pooling layer. To enhance feature selection, the study proposed an improved Quantum Theory-based Marine Predator Optimization algorithm (QTbMPA) for selecting the most relevant features from both networks. The selected features were fused using a serial-based approach and classified with neural network classifiers. Experimental results on an augmented Figshare dataset demonstrated an accuracy of 99.80%, with a sensitivity of 99.83% and a precision of 99.83%, highlighting the effectiveness of the proposed framework.

Rasheed et al. [34] investigated the classification of brain tumors using deep learning techniques on MRI images. The study leveraged convolutional neural networks (CNNs) combined with a hybrid attention mechanism to classify glioma, meningioma, pituitary tumors, and non-tumor cases. Benchmark datasets were utilized to evaluate the proposed method against established pre-trained models such as Xception, ResNet50V2, DenseNet201, ResNet101V2, and DenseNet169. Experimental results demonstrated superior performance, achieving a classification accuracy of 98.33%, with a precision of 98.30%, recall of 98.30%, and an F1-score of 98.20%. These findings highlight the effectiveness of the hybrid attention-based CNN approach in improving brain tumor classification accuracy and generalization, making it a valuable tool for medical diagnostics.

3 Artificial Intelligence

Artificial intelligence is the branch of science that enables human intelligence to fulfill various tasks by imitating human intelligence by machine. It is aimed to realize the learning process carried out by cells called neurons in humans by computers based on a mathematical basis. Today, it is used extensively in strategic games such as chess and go, speech recognition, driverless cars, personal assistants and many other fields. Although artificial intelligence is a general term, it is divided into many sub-branches such as Artificial Neural Networks, Machine Learning, Image Processing, Natural Language Processing, Expert Systems.

3.1 Machine Learning

Machine learning is a sub-branch of Artificial Intelligence. Machine learning involves computers discovering how to perform tasks without being explicitly programmed to do so. It involves computers learning from data provided to them to perform certain tasks. For simple tasks assigned to computers, it is possible to program algorithms that tell the machine how to perform all the steps needed to solve the problem at hand; no learning is required on the computer's side. For more advanced tasks, it can be difficult for a human to manually create the necessary algorithms. In practice, it may be more effective to help the machine develop its own algorithm, rather than having human programmers determine every necessary step.

The discipline of machine learning uses various approaches to teach computers to perform tasks for which there is no fully satisfactory algorithm. In cases where there are many possible answers, one approach is to label some of the correct answers as valid. This can then be used

as training data to develop the algorithm(s) the computer uses to find the correct answers. For example, the MNIST dataset of handwritten digits is often used to train the system for the task of numeric character recognition.

3.2 Deep Learning

Deep Learning, a subfield of machine learning, is a field of study that covers artificial neural networks and similar machine learning algorithms that contain one or more hidden layers.

In other words, it is the use of at least one artificial neural network (ANN) and many algorithms to obtain new data from the data at hand.

Deep learning can be performed supervised, semi-supervised or unsupervised. Deep neural networks have also shown successful results with a reinforcement learning approach. Neural networks are inspired by information processing and distributed communication nodes in biological systems. Neural networks have several differences from biological brains. In particular, neural networks tend to be static and symbolic, whereas the biological brain of most living organisms is dynamic (plastic) and analog.

4 Materials and Methods

The training process consists of four stages: preprocessing the images, splitting the data into training and test sets, utilizing CNN models for feature extraction by structuring their layers, and selecting the optimal hyperparameters.

The dataset contains images with a resolution of 1615×840. Since the input sizes of the CNN models used for feature extraction vary, the images have been resized to match the respective input dimensions of the models as a preprocessing step. No additional processing has been applied. Subsequently, the data has been divided into two parts: training and test sets.

The training data has been used for feature extraction to select the best parameters, subsequently, machine learning models have been trained using these optimal parameters and tested with test data. In the study, feature extraction has been performed using 4 CNN (Convolutional Neural Network) models, classification has been performed with SVM and XGBoost machine learning algorithms and performance comparison has been conducted. Figure 1 shows the flow diagram of the model.

While the basic architecture of the CNN models used in the study is kept constant, the upper layers have been removed and a GlobalAveragePooling2D layer has been added. The GlobalAveragePooling2D layer summarizes the features obtained after convolution and pooling layers and provides more general feature extraction by reducing the number of features. This results in computational and time savings.

Auxiliary functions have been employed to determine the parameters that the machine learning models achieve optimal performance. For parameter optimization, the RandomSearchCV class from the sklearn library, a Python programming language library, has been utilized. The RandomSearchCV class is used to optimize the parameters used in machine learning models. Manually finding the optimal parameter set and combination that yields the best performance in machine learning models is a time-consuming process. The RandomSearchCV class divides the available dataset into parts with the K-Fold method and evaluates the performance of the machine learning model by making random selections from the list of parameters to be optimized and displays the parameter group with the highest performance. In this study, RandomSearchCV has been used to find the most optimal parameters for the features extracted from CNN models to work effectively with machine learning algorithms. Configured with 5 K-fold and 20 iterations, a total of 100 iterations have been performed, and the model has been trained using the optimal values found by the RandomSearchCV class, with the results being reported.

Another parameter optimization class, GridSearchCV, has not been preferred in this study as it tries all possible combinations and causes a long processing time due to the large size of the attributes obtained from the dataset used in the study.

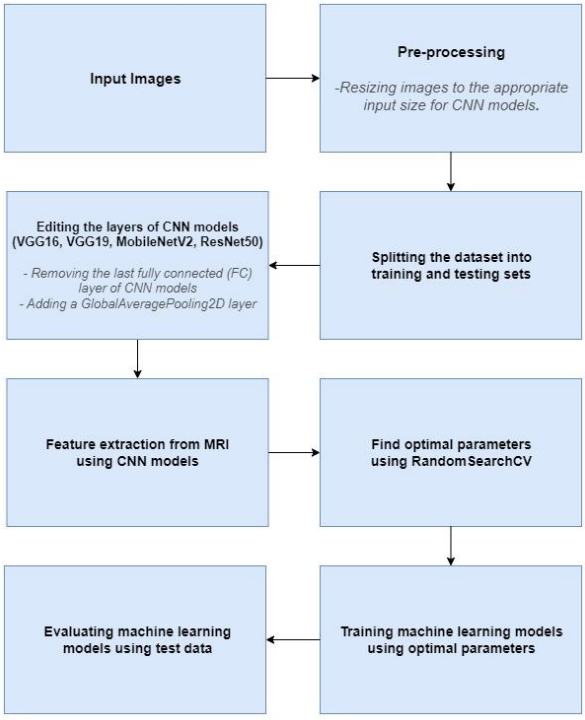

Figure 1. Model training flowchart.

4.1 Dataset

The dataset used in the study has been obtained from the Kaggle [35]. The dataset has 4 classes as glioma, meningioma, pituitary and non-tumor and contains 7023 images in jpeg format. Table 1 shows the distribution of classes in the dataset. Figure 2 shows sample images from the dataset. This dataset has been created by merging figshare, SARTAJ, and Br35H datasets. Since the images are of varying sizes, so all images have been standardized before training. The dataset has been split into two parts, 80% training and 20% testing.

Table 1. Number of classes in the dataset.

Glioma	Meningioma	Pituitary	No Tumor	Total
1621	1645	1757	2000	7023

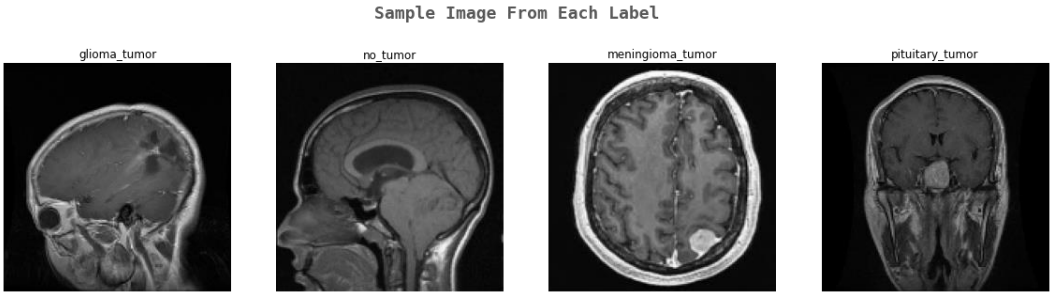

Figure 2. Sample MRI images from the dataset.

4.2 Evaluation Metrics

Accuracy, Sensitivity, Precision and F1-score metrics are used to evaluate the models after training. These metrics are calculated based on the values of TP (Positive True), TN (Negative True), FP (Positive False), FN (Negative False).

A confusion matrix is a table used to evaluate the performance of a classification model. It is used to evaluate two classes of cases (positive and negative). In particular, it is the basis for calculating performance metrics such as recall and precision. Considering a two-class case, it includes four main categories:

- True Positive (TP): The case where true positive instances are correctly identified. Instances that the model labels as positive are indeed positive.
- True Negative (TN): The case where true negative instances are correctly identified. Instances that the model labels as negative are indeed negative.
- False Positive (FP): The case where true negative instances are incorrectly identified as positive. The model labeled samples as positive when they should have been negative.
- False Negative (FN): The case where true positive instances are incorrectly identified as negative. The model labeled samples that should have been positive as negative.

The confusion matrix is used to gain a more detailed understand of which classes the model performs better or worse. It also provides as the foundation for calculating metrics such as the F1-score.

Recall measures the extent to which the model misses true positives (true positives). In particular, it is used for situations that are important to detect. For example, in a model that detects a disease, a low recall value may indicate that some of the people who are actually sick are not detected. Recall ranges from 0 to 1, with 1 representing an ideal situation.

Precision measures how many of the samples that the model labels as positive are actually positive. It is used when it is important to limit false positives. For example, in a spam email filtering application, high precision prevents the user from mistakenly marking important emails as spam. Precision ranges from 0 to 1, with 1 representing an ideal situation.

The F1-score provides a balance by taking the harmonic mean of recall and precision. This balance is particularly important when the data is unbalanced or when false positive and false negative errors are of equal importance. The F1-score is useful when the model needs to balance both false positive and false negative errors. The F1-score value ranges from 0 to 1, with 1 representing an ideal situation.

4.3 Support Vector Machines (SVM)

Support Vector Machines are a versatile and powerful machine learning model. It can perform both linear or non-linear classification, and can also be applied in regression tasks. It is mostly used in classification processes. It is one of the most popular models in machine learning. It is highly suitable for classifying small or medium-sized complex data sets. An SVM uses a hyperplane to separate points on a plane. This hyperplane can be a point, a line, a 3D plane or more multidimensional planes. The primary objective of the SVM model is to separate these points while maximizing the margin of the decision boundary. It uses a set of parameters to achieve this.

4.3.1 Kernel Parameter

SVM does not only work with 2-dimensional data. It can also work with multidimensional data using various kernels, a technique known as the kernel trick. For example, data that cannot be separated in 2 dimensions can be transformed into separable data by mapping it to 3 or more dimensions. The SVM classifier in the scikit-learn library used in the study uses 4 kernels: Linear Kernel, Poly Kernel, RBF Kernel and Sigmoid Kernel.

Linear Kernel aims to separate classes on a 2D plane with a single linear boundary. The area between two classes is called margin. The higher the margin value, the better and more accurate the performance of the classifier. The mathematical representation of the Linear Kernel is given in Equation 1. It computes the dot product of two feature vectors, x_1 and x_2. Here, x_1 and x_2 represent data points in the feature space. The dot product measures their similarity in a linear space. A higher dot product value indicates that the two data points are more similar.

Poly Kernel (Polynomial Kernel) is used to separate data that cannot be explained in 2 dimensions. The higher the polynomial degree, the better the fit to the data. High polynomial degree can lead to over-fitting. The

mathematical representation of the Poly Kernel is given in Equation 2. It takes two feature vectors, x_1 and x_2, and applies a polynomial transformation. Here, x_1 and x_2 represent data points in the feature space. The parameter γ scales the dot product, affecting the influence of input features. The term r is a constant that shifts the polynomial function, and d determines the polynomial degree. A larger d increases complexity, making the model more flexible but also more prone to overfitting.

RBF (Radial Base Function), another kernel used to separate high-dimensional data, transforms the attributes by calculating the distance to the specified reference points and performs re-separation in this way. The mathematical representation of the RBF Kernel is given in Equation 3. It calculates the similarity between two feature vectors, x_1 and x_2, based on their squared Euclidean distance. The parameter γ controls the spread of the function, determining how much influence each data point has. A higher γ makes the model more sensitive to nearby points, capturing fine details but increasing the risk of overfitting, while a lower γ results in a smoother decision boundary.

Sigmoid kernel, although adapted for SVM, is an imported function from Artificial Neural Networks. By its nature, it is highly error-prone and is not widely used in practice. It is disadvantageous in that positive values are difficult to obtain and may cause uneven updating of weights in backpropagation. Although it is not commonly preferred in research, it has been included in this study to assess its impact. The mathematical representation of the Sigmoid Kernel is given in Equation 4. It calculates the similarity between two feature vectors, x_1 and x_2, by applying the hyperbolic tangent function to their dot product. The parameter γ controls the influence of the dot product, and r is a constant that shifts the output. The function is sensitive to changes in γ, with higher values leading to a more complex decision boundary, while r adjusts the overall scaling of the function.

$$K(x_1, x_2) = x_1^T x_2 \quad (1)$$

$$K(x_1, x_2) = (\gamma \cdot x_1^T x_2 + r)^d \quad (2)$$

$$K(x_1, x_2) = \exp(-\gamma \cdot \|x_1 - x_2\|^2) \quad (3)$$

$$K(x_1, x_2) = \tanh(\gamma \cdot x_1^T x_2 + r) \quad (4)$$

4.3.2 C Parameter

The goal of the SVM model is to find the optimal hyperplane. Maintaining the decision boundary as wide as possible while finding the hyperplane is crucial for the generalization performance of the model. However, if one insists on keeping the attributes on the correct side and outside this hyperplane with the calculated hyperplane, this is called hard classification (hard margin). However, allowing a few attributes that can be missed is called soft margin. The parameter C can be used to help determine this margin when building the SVM model. This parameter is effective if a Linear Kernel is used for the SVM classifier. The lower the C parameter is chosen, the higher the level at which the model is able to sift through these outlier features. So a low C value leads to soft classification, while a high C value leads to hard classification. A high C value can lead to overfitting of the model

As shown in Figure 3, when the C parameter is set to a low value, outliers are more tolerant of being outside the hyperplane and soft classification is performed, which leads to a wider calculation of the decision boundary of the two classes separated by the hyperplane. As can be seen in the same figure, if the C parameter is set to a high value, the area separated by the hyperplane is narrower and has the effect of covering more attributes. This situation can lead to overfitting and may cause a decrease in generalization performance. To select the optimal C value, hyperparameter tuning techniques should be employed.

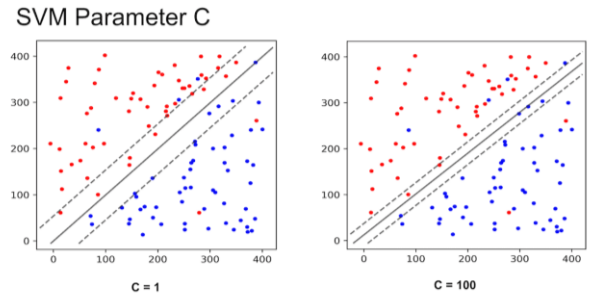

Figure 3. The effect of C parameter in SVM Classifier.

4.3.3 Gamma Parameter

The gamma parameter, used for class separation with the Radial Basis Function, adjust the degree of curvature of the bell curve generated by the function. A high gamma value makes the bell curve more curved, which can cause the model to overfit. A low gamma value makes the bell curve flatter, but can lead to under-learning. When used in combination with a C value, the performance of class separation can be improved. If an SVM model utilizing the RBF Kernel is over-fitting, it is recommended to decrease the gamma value and if it is under-fitting, it is recommended to increase the gamma value. Figure 4 shows the effect of C and gamma parameters on class distinction.

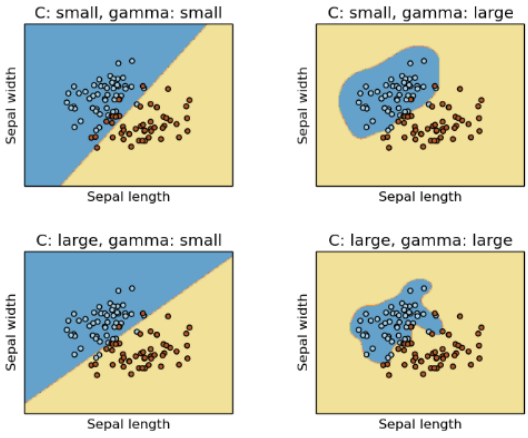

Figure 4. Effect of C and Gamma values on class distinction.

4.4 XGBoost

XGBoost is a machine learning technique built upon the Gradient Boosting algorithm, which is based on the boosting algorithm, one of the ensemble algorithms. The boosting algorithm consists of a set of algorithms that try to optimize an ensemble of a set of weak learners. Each model aims to make better predictions based on the shortcomings of the previous model. Thus, strong learners are formed from multiple weak learners. Decision trees are also utilized as weak learners in the structure of the system.

Gradient Boosting is a boosting algorithm that seeks to minimize errors through gradient descent. It calculates the gradient of the loss function at each iteration. XGBoost also includes L1 (Lasso) and L2 (Ridge) regularization techniques to prevent overfitting. By controlling the complexity of the trees, these techniques enhance the generalizability of the model.

XGBoost is used in many tasks such as regression and classification. It is one of the most widely popular machine learning algorithms today.

4.4.1 Max Depth Parameter

Depending on the type of problem and dataset, more complex relationships can be learned by increasing the depth of the decision trees used in the XGBoost algorithm. However, deeper trees may lead to over-fitting. Therefore, random or grid search methods can be employed for optimal selection.

4.4.2 Learning Rate

In the XGBoost algorithm, the prediction of each tree is sequentially added to the overall sum. By weighting the contribution of each tree, the learning rate can be slowed down. This is called shrinkage. The XGBoost algorithm uses a learning rate coefficient to achieve this. In general, choosing a low learning rate allows the model to learn more robustly, although training may take longer.

4.4.3 Subsample

During training, a random subset of the dataset is selected for training each tree. By training the model on different subsets, this approach helps prevent overfitting.

4.4.4 N Estimators

Each tree used in the XGBoost algorithm is referred to as an estimator. Predicting the optimal number of trees, can be challenging. A small number of estimators can lead to under-fitting, while a large number can lead to over-fitting. The ideal number of estimators can only be chosen using techniques such as random or grid search. The search is terminated when there is no improvement in the performance of the model after a certain point. Figure 5 shows the effect of the N Estimators parameter on the model performance.

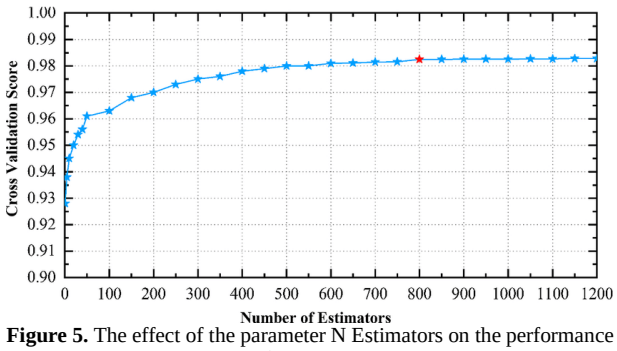

Figure 5. The effect of the parameter N Estimators on the performance of the model.

4.5 RandomSearchCV

When models are created, they are initialized with various hyperparameters. The models operate, learn, and make predictions based on these hyperparameters. It is clear that hyperparameters are present at every stage of the models, so the correct choice of parameters will directly affect the performance of the model. The right choice of hyperparameters depends on the problem and the data set. A set of hyperparameters that works well for one model may not be appropriate for another.

For hyperparameters that vary from model to model, various methods have been developed to select the most appropriate hyperparameter. RandomSearchCV is one of these methods. The RandomSearchCV class is included in the scikit-learn library. The purpose of this class is to test the performance of the model by randomly selecting the hyperparameters of the model over a specified range. In this way, it is aimed to obtain suitable values in a short time since the entire range specified for the hyperparameters is not scanned. RandomSearchCV utilizes the cross-validation method, where instead of using the entire dataset, a portion of the dataset used for training, and the remaining part for testing. Thus, it more accurately measures the fit of the parameter set for the model.

Unlike the RandomSearchCV class, the GridSearchCV class searches over the entire specified range. This requires a long time to select the suitable parameters. Since the GridSearchCV class searches over all possibilities, it is considered the most reliable method. However, considering the costs and time constraints, the RandomSearchCV class is more advantageous. In this study, RandomSearchCV class has been employed to select suitable hyperparameters for the machine learning models used in this study.

5 Results and Discussion

In this study, Accuracy, Sensitivity, Precision and F1-score metrics, which are used to evaluate the learning success of a model, are taken into consideration. First, features have been extracted from the 4 CNN models respectively and these features have been given to SVM and XGBoost machine learning models separately. In SVM model training, the optimum values of C and gamma parameters have been determined with RandomSearchCV. At the same time, 4 kernels used in SVM models have been tested respectively to determine the kernel with the best performance. The optimum values

of max_depth, learning rate, subsample, n_estimators parameters of the XGBoost model have been also determined by RandomSearchCV. The results of feature extraction from CNN models and subsequent classification with machine learning models are listed in Table 2. Following the evaluation of the models with the test data, cofusion matrices have been generated. Figure 6, 7, 8 and 9 illustrate the cofusion matrices produced as a result of testing the features obtained separately from 4 CNN models in the SVM model according to the kernel selection. Similarly, the cofusion matrices generated by testing the features obtained from 4 CNN models in the XGBoost model are given in Figure 10, 11, 12 and 13.

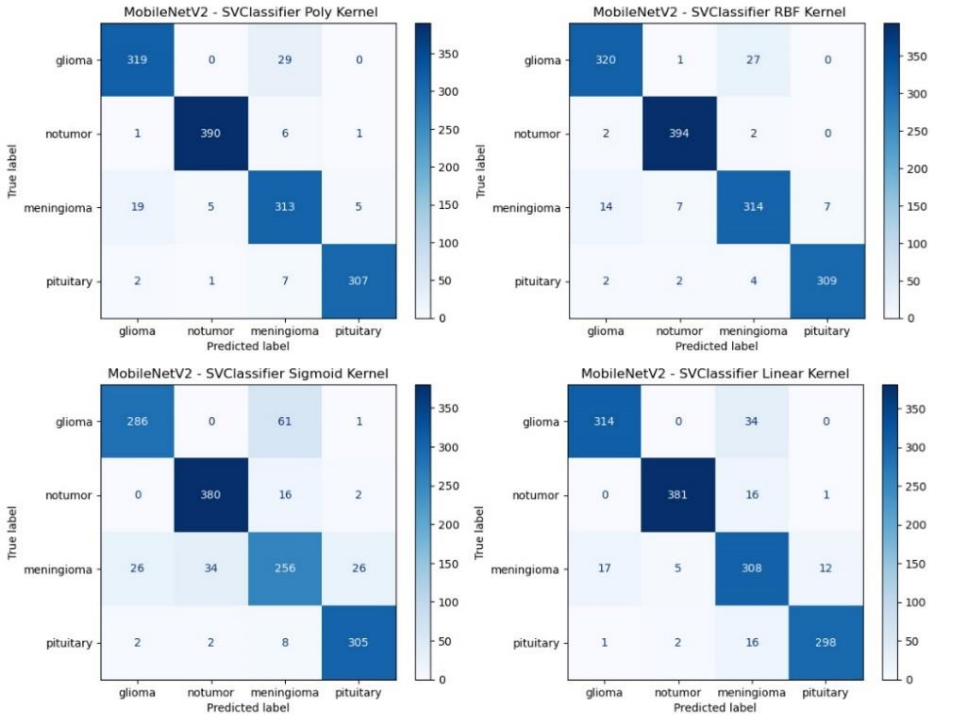

Figure 6. Cofusion Matrices of MobileNetV2 SVM Classifier.

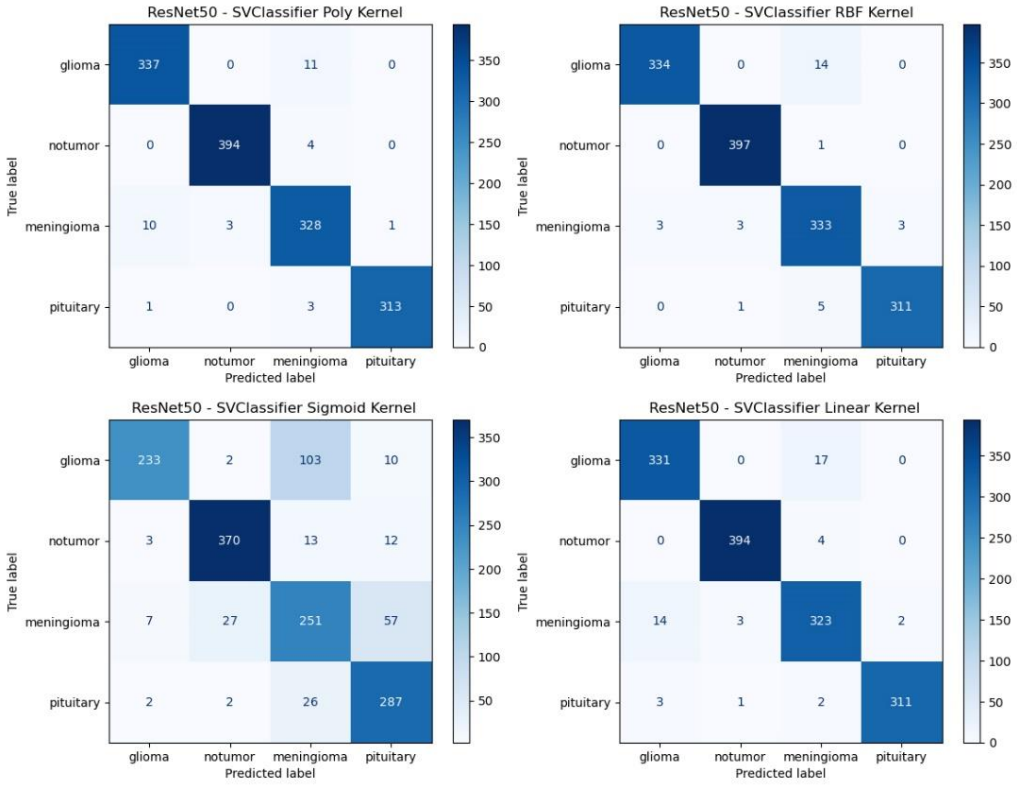

Figure 7. Cofusion Matrices of ResNet50 SVM Classifier.

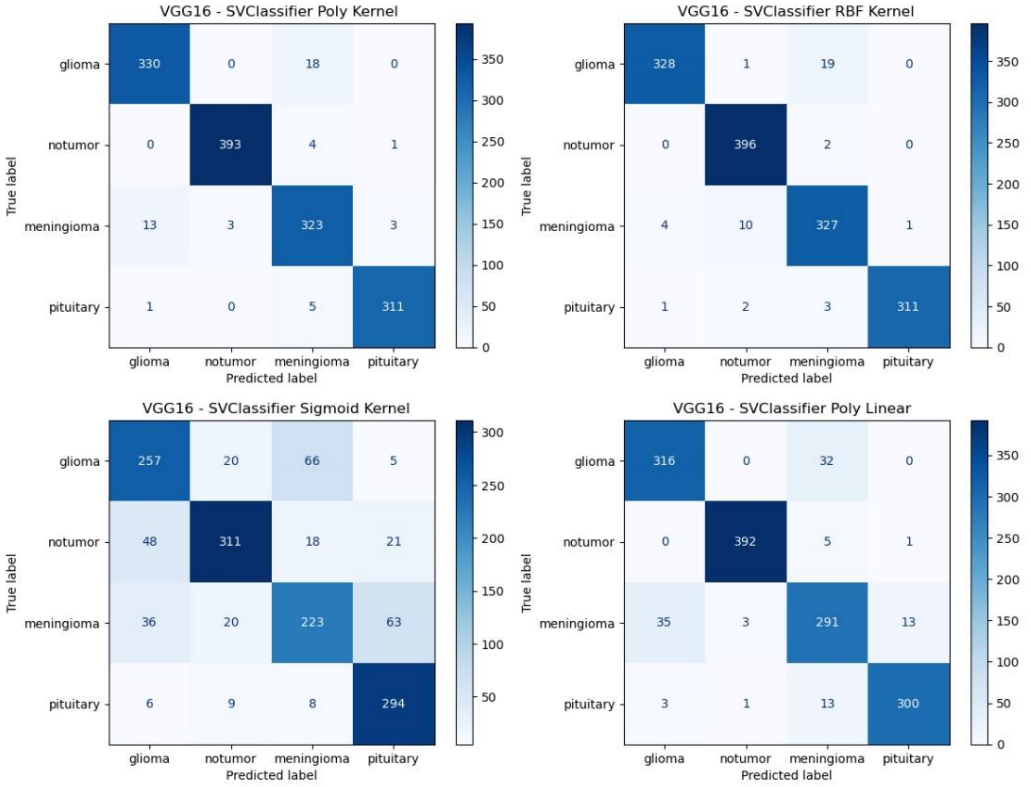

Figure 8. Cofusion Matrices of VGG16 SVM Classifier.

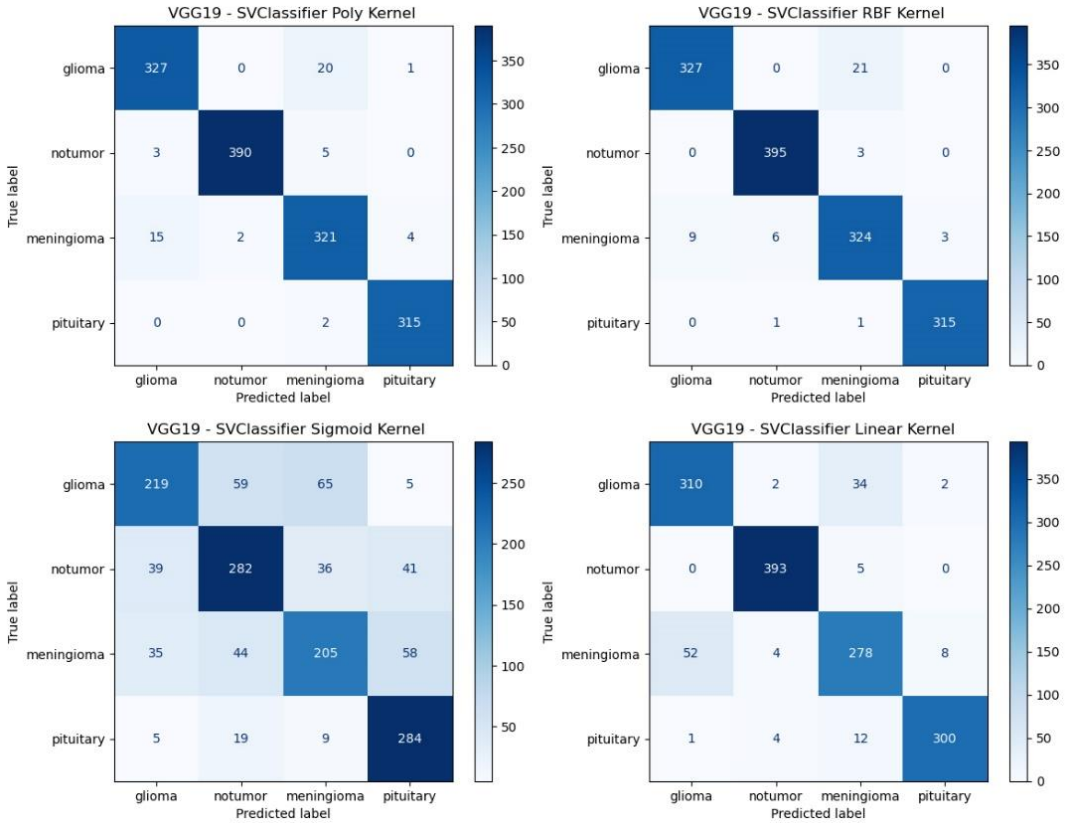

Figure 9. Cofusion Matrices of VGG19 SVM Classifier.

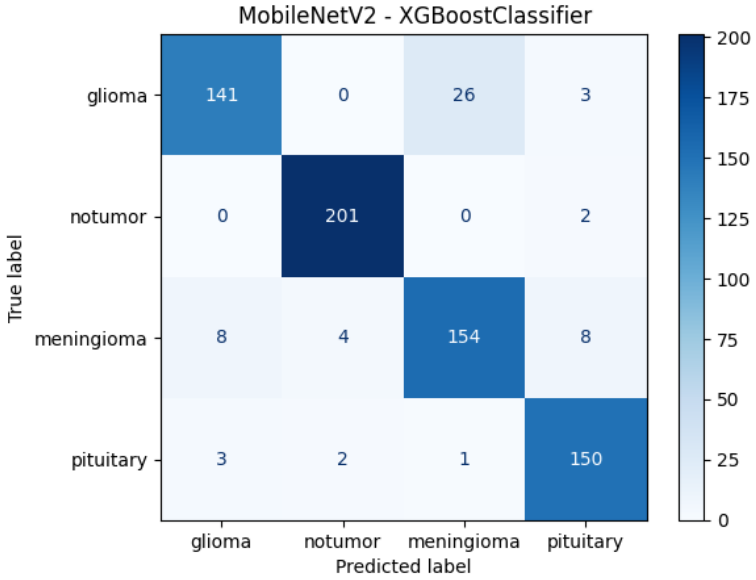

Figure 10. Confusion Matrices of MobileNetV2 XGBoost Classifier.

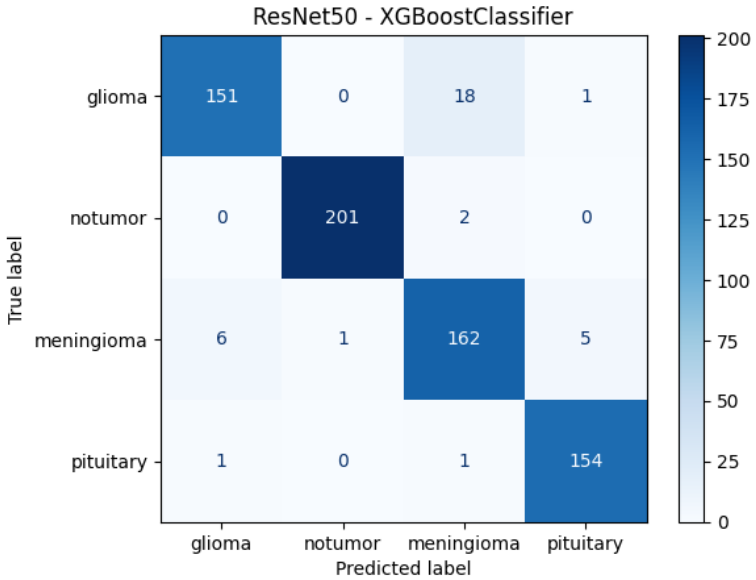

Figure 11. Confusion Matrices of ResNet50 XGBoost Classifier.

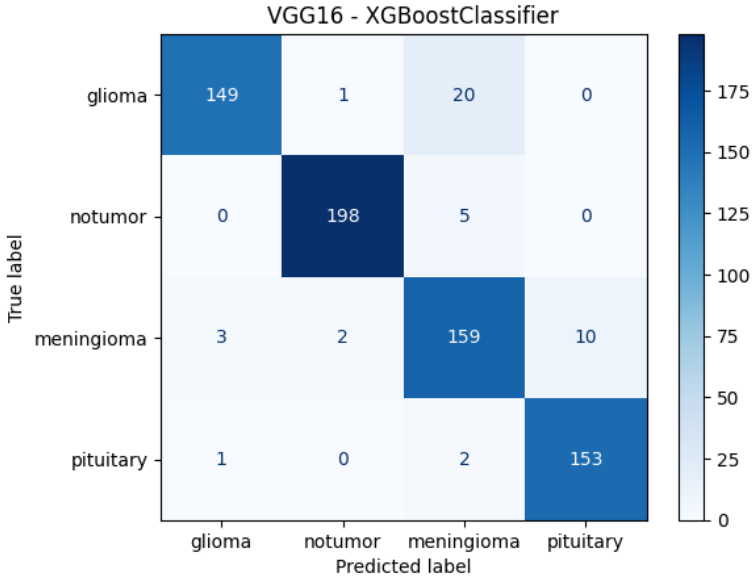

Figure 12. Confusion Matrices of VGG16 XGBoost Classifier.

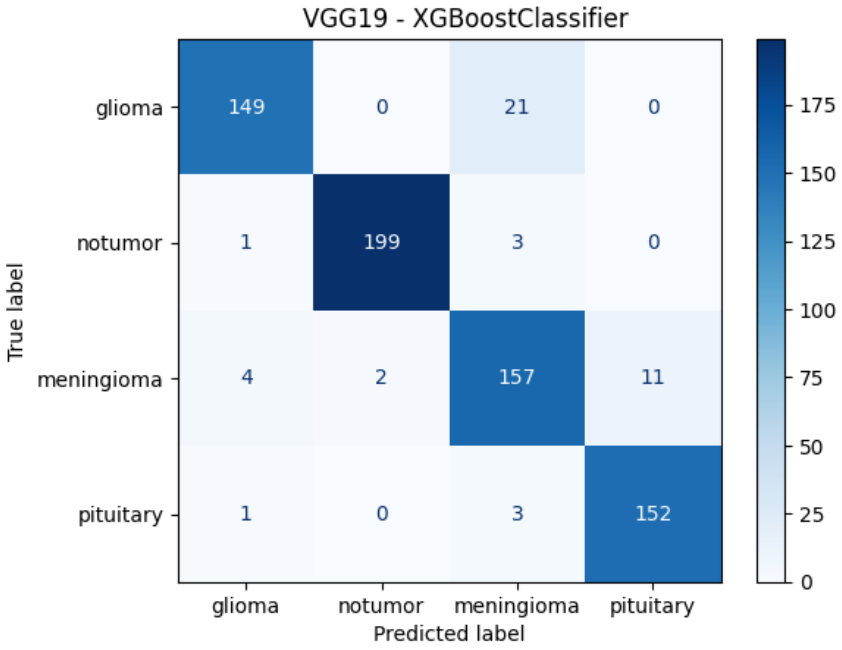

Figure 13. Confusion Matrices of VGG19 XGBoost Classifier.

Table 2. Test results of CNN models and machine learning models after training.

Method	Accuracy	Precision	Recall	F1-Score
CNN (MobileNetV2) + SVM (Poly Kernel)	0.9459	0.9466	0.9459	0.9461
CNN (MobileNetV2) + SVM (RBF Kernel)	0.9688	0.9517	0.9516	0.9515
CNN (MobileNetV2) + SVM (Sigmoid Kernel)	0.8733	0.8731	0.8733	0.8724
CNN (MobileNetV2) + SVM (Linear Kernel)	0.9259	0.9291	0.9260	0.9269
CNN (ResNet50) + SVM (Poly Kernel)	0.9765	0.9766	0.9765	0.9766
CNN (ResNet50) + SVM (RBF Kernel)	**0.9786**	**0.9790**	**0.9786**	**0.9787**
CNN (ResNet50) + SVM (Sigmoid Kernel)	0.8120	0.8293	0.8121	0.8129
CNN (ResNet50) + SVM (Linear Kernel)	0.9672	0.9674	0.9673	0.9673
CNN (VGG16) + SVM (Poly Kernel)	0.9658	0.9661	0.9658	0.9659
CNN (VGG16) + SVM (RBF Kernel)	0.9693	0.9699	0.9694	0.9694
CNN (VGG16) + SVM (Sigmoid Kernel)	0.7722	0.7737	0.7722	0.7704
CNN (VGG16) + SVM (Linear Kernel)	0.9245	0.9248	0.9246	0.9247
CNN (VGG19) + SVM (Poly Kernel)	0.9629	0.9632	0.9630	0.9631
CNN (VGG19) + SVM (RBF Kernel)	0.9686	0.9689	0.9687	0.9687
CNN (VGG19) + SVM (Sigmoid Kernel)	0.7046	0.7033	0.7046	0.7008
CNN (VGG19) + SVM (Linear Kernel)	0.9117	0.9118	0.9117	0.9116
CNN (MobileNetV2) + XGBoost	0.9189	0.9288	0.9275	0.9270
CNN (ResNet50) + XGBoost	0.9502	0.9486	0.9474	0.9474
CNN (VGG16) + XGBoost	0.9374	0.9400	0.9388	0.9389
CNN (VGG19) + XGBoost	0.9345	0.9382	0.9360	0.9361

As a result of the evaluation of the results, it has been observed that the best results have been obtained with the SVM classifier with the RBF kernel of the features obtained from the ResNet50 CNN model with 97.87% F1-score performance.

When evaluating the features extracted from the four CNN models, it has been observed that the best performances have been achieved across all CNN models utilizing the RBF kernel. Conversely, the performance with the Sigmoid kernel has been consistently lower across all CNN models.

Through the application of transfer learning, features from the ResNet50 model have been obtained quickly and effectively and have been classified with high accuracy by the SVM classifier. The best performance for the XGBoost classifier is 94.74% with the features obtained from the ResNet50 CNN model.

Table 3 presents a comparison of this study with other studies in the literature that utilize the same dataset. The results indicate that the classification performance achieved using SVM in this study demonstrates a higher accuracy rate compared to studies that employ CNN for feature extraction followed by classification with machine learning algorithms. However, studies that perform end-to-end feature extraction and classification using CNN models have reported superior performance.

Table 3. Comparison of results with other studies that utilized the same dataset.

Researchs	Methods	Evaluation Metrics	Rates
Karamehić et al. [1] (2023)	VGG16 CNN model	Accuracy	96.9%
Remzan et al. [2] (2023)	CNN models (ResNet50, VGG19, EfficientNetV2B1) for feature extraction, MLP classifier	Accuracy	96.67% (MLP classification)
Pal et al. [3] (2023)	VGG16 CNN model	Accuracy	92% (Overall) 90% (Meningioma) 91% (Pituitary) 93% (Glioma) 89% (Non-tumor)
Guzmán et al. [22] (2023)	CNN models (ResNet50, InceptionV3, InceptionResNetV2, Xception, MobileNetV2, EfficientNetB0)	Accuracy	97.12% (InceptionV3)
Bahya et al. [27] (2023)	FFT, K-means clustering, Tamura texture analysis, Gradient Boosting (GB), SVM, AdaBoost, SVM	Accuracy	Normal/Abnormal Classification: 92.6% (Gradient Boosting) 86% (ADA) 82% (SVM) Tumor Classification: GB: 67.9% SVM: 65.3% ADA: 59.6%
Dewan et al. [28] (2023)	GLCM for feature extraction, KNN, SVM, Decision Tree, Naïve Bayes, Logistic Regression, Random Forest for classification	Accuracy	91.04% (Random Forest)
Akter et al. [29] (2024)	CNN for classification, U-Net for segmentation, Benchmark datasets	Accuracy, Recall, Precision, AUC	98.7% (Merged dataset), 98.8% (with segmentation)
N. et al. [30] (2023)	CNN, Transfer learning (VGG16, AlexNet, ResNet50), Optimizers (Adam, AdaDelta, SGD)	Accuracy	99.83% (VGG16 with AdaDelta)
Filatov et al. [31] (2022)	CNN (ResNet50, EfficientNetB1, EfficientNetB7, EfficientNetV2B1)	Accuracy	EfficientNetB1: 87.67% (training) 89.55% (validation)
Islam et al. [32] (2023)	Transfer learning (InceptionV3, VGG19, DenseNet121, MobileNet)	Accuracy	99.60% (MobileNet)
Ullah et al. [33] (2024)	Sparse autoencoder network, Bayesian optimization, Quantum Theory-based Marine Predator Optimization (QTbMPA)	Accuracy, Sensitivity, Precision	99.80% (Accuracy) 99.83% (Sensitivity) 99.83% (Precision)
Rasheed et al. [34] (2024)	CNN with hybrid attention mechanism, Benchmark datasets	Accuracy, Precision, Recall, F1-score	98.33% (Accuracy) 98.30% (Precision) 98.30% (Recall) 98.20% (F1-score)
This study (2024)	Feature extraction with VGG16, VGG19, ResNet50 and MobileNetV2, classification with SVM and XGBoost	Accuracy, Precision, Recall, F1-score	XGBoost: 95.02% (Accuracy) 94.86% (Precision) 94.74% (Recall) 94.74% (F1-score) SVM: 97.86% (Accuracy) 97.90% (Precision) 97.86% (Recall) 97.87% (F1-Score)

In this study, the GlobalAveragePooling2D layer added to the final layer of the CNN models has reduced both computational and time costs. All parameters have been tested using the RandomSearchCV class, and both positive and negative outcomes have been reported. Thus, the effects of the added layers and auxiliary classes have been evaluated through the conducted tests.

No augmentation has been applied in this study, and no enhancements have been made to the images. Additionally, CNN models have been used for feature extraction without fine-tuning on MRI images. In future studies, it is anticipated that image enhancements, fine-tuning of CNN models using MRI images for more effective feature extraction, and the use of data augmentation may improve the performance of machine learning algorithms

Declaration

Ethics committee approval is not required.

References

[1] Karamehić, S., & Jukić, S. (2023). Brain tumor detection and classification using VGG16 Deep learning algorithm and Python Imaging Library. Bioengineering Studies, 4(2), 1-13.

[2] Remzan, N., Hachimi, Y. E., Tahiry, K., & Farchi, A. (2023). Ensemble learning based-features extraction for brain MR images classification with machine learning classifiers. *Multimedia Tools and Applications.*

[3] Pal, S. S., Raymahapatra, P., Paul, S., Dolui, S., Chaudhuri, A. K., & Das, S. (2023). A novel brain tumor classification model using machine learning techniques. *International Journal of Engineering Technology and Management Sciences, 7*(2), 87–98.

[4] Bohra, M., & Gupta, S. (2022). Pre-trained CNN models and machine learning techniques for brain tumor analysis. In *2022 2nd International Conference on Emerging Frontiers in Electrical and Electronic Technologies (ICEFEET)* (pp. 1–6).

[5] Hong, J. (2021). Feasibility evaluation of brain tumor magnetic resonance imaging classification using convolutional neural network model. *Journal of the Korean Society of MR Technology, 31*(1), 17–23.

[6] Latif, G., Bashar, A., Iskandar, D. N. F. A., Mohammad, N., Brahim, G. B., & Alghazo, J. M. (2023). Multiclass tumor identification using combined texture and statistical features. *Medical & Biological Engineering & Computing, 61*(1), 45–59.

[7] Kibriya, H., Masood, M., Nawaz, M., Rafique, R., & Rehman, S. (2021). Multiclass brain tumor classification using convolutional neural network and support vector machine. *Conference Proceedings*, 1–4.

[8] Tandel, G. S., Balestrieri, A., Jujaray, T., Khanna, N. N., Saba, L., & Suri, J. S. (2020). Multiclass magnetic resonance imaging brain tumor classification using artificial intelligence paradigm. *Computers in Biology and Medicine, 122*, 103804.

[9] Kumar, R., Kakarla, J., Isunuri, B., & Singh, M. (2021). Multi-class brain tumor classification using residual network and global average pooling. *Multimedia Tools and Applications, 80.*

[10] Gurkahraman, K., & Karakış, R. (2021). Veri çoğaltma kullanılarak derin öğrenme ile beyin tümörlerinin sınıflandırılması. *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi, 36*(2), 997–1012.

[11] Tasci, B. (2022). Beyin MR görüntülerinden mRMR tabanlı beyin tümörlerinin sınıflandırması. *Journal of Scientific Reports-B*(006), 1–9.

[12] Uysal, F., & Erkan, M. (2023). Evrişimsel sinir ağları temelli derin öğrenme modelleri kullanılarak beyin tümörü manyetik rezonans görüntülerinin sınıflandırılması. *EMO Bilimsel Dergi, 13*(2), 19–27.

[13] Demir, K., Arı, B., & Demir, F. (2023). Detection of brain tumor with a pre-trained deep learning model based on feature selection using MR images. *Firat University Journal of Experimental and Computational Engineering, 2*(1), 23–31.

[14] Aslan, M. (2022). Derin öğrenme tabanlı otomatik beyin tümör tespiti. *Fırat Üniversitesi Mühendislik Bilimleri Dergisi, 34*(1), 399–407.

[15] Paul, J. S., Plassard, A. J., Landman, B. A., & Fabbri, D. (2017). Deep learning for brain tumor classification. In A. Krol & B. Gimi (Eds.), *Medical Imaging 2017: Biomedical Applications in Molecular, Structural, and Functional Imaging* (p. 1013710). SPIE.

[16] Srinivas, B., & Rao, G. S. (2019). A hybrid CNN-KNN model for MRI brain tumor classification. *International Journal of Recent Technology and Engineering, 8*(2), 5230–5235..

[17] Jayade, S., Ingole, D. T., & Ingole, M. D. (2019). MRI brain tumor classification using hybrid classifier. In *2019 International Conference on Innovative Trends and Advances in Engineering and Technology (ICITAET)* (pp. 201–205).

[18] Ayadi, W., Charfi, I., Elhamzi, W., & Atri, M. (2022). Brain tumor classification based on hybrid approach. *The Visual Computer, 38*(1), 107–117.

[19] Amin, J., Sharif, M., Gul, N., Yasmin, M., & Shad, S. A. (2020). Brain tumor classification based on DWT fusion of MRI sequences using convolutional neural network. *Pattern Recognition Letters, 129*, 115–122.

[20] Shahin, A. I., Aly, W., & Aly, S. (2023). MBTFCN: A novel modular fully convolutional network for MRI brain tumor multi-classification. *Expert Systems with Applications, 212*, 118776.

[21] Kang, J., Ullah, Z., & Gwak, J. (2021). MRI-based brain tumor classification using ensemble of deep features and machine learning classifiers. *Sensors, 21*(6), Article 6.

[22] M Gómez-Guzmán, M. A., et al. (2023). Classifying brain tumors on magnetic resonance imaging by using convolutional neural networks. *Electronics, 12*(4).

[23] Tiwari, P., et al. (2022). CNN based multiclass brain tumor detection using medical imaging. *Computational Intelligence and Neuroscience, 2022*, 1830010.

[24] Aamir, M., et al. (2022). A deep learning approach for brain tumor classification using MRI images. *Computers and Electrical Engineering, 101*, 108105.

[25] Saleh, A., Sukaik, R., & Abu-Naser, S. S. (2020). Brain tumor classification using deep learning. In *2020 International Conference on Assistive and Rehabilitation Technologies (iCareTech)* (pp. 131–136).

[26] Muhammad, L. J., Badi, I., Haruna, A. A., Mohammed, I. A., & Dada, O. S. (2022). Deep learning models for classification of brain tumor with magnetic resonance imaging images dataset. In K. Raza (Ed.), *Computational Intelligence in Oncology* (pp. 159–176). Springer Singapore.

[27] Bahya, T. M., & Hussein, N. (2023). Machine learning techniques to classify brain tumor. In *2023 6th International Conference on Engineering Technology and its Applications (IICETA)* (pp. 609–614).

[28] Dewan, J., Thepade, S., Deshmukh, P., Deshmukh, S., Katpale, P., & Gandole, K. (2023). Brain tumor type identification from MR images using texture features and machine learning techniques. *Research Square.*

[29] Akter, A., et al. (2024). Robust clinical applicable CNN and U-Net based algorithm for MRI classification and segmentation for brain tumor. *Expert Systems with Applications, 238*, 122347.

[30] V. N., S. K., Kodipalli, A., Rao, T., & K. S. (2023). Comparative study of customized CNN model and transfer learning for brain tumor classification. In *2023 International Conference on Network, Multimedia and Information Technology (NMITCON)* (pp. 1–6).

[31] Filatov, D., & Yar, G. N. A. H. (2022). Brain tumor diagnosis and classification via pre-trained convolutional neural networks. *arXiv preprint arXiv:2208.00768*.

[32] Islam, M. M., Barua, P., Rahman, M., Ahammed, T., Akter, L., & Uddin, J. (2023). Transfer learning architectures with fine-tuning for brain tumor classification using magnetic resonance imaging. *Healthcare Analytics, 4*, 100270.

[33] Ullah, M. S., Khan, M. A., Masood, A., Mzoughi, O., Saidani, O., & Alturki, N. (2024). Brain tumor classification from MRI scans: A framework of hybrid deep learning model with Bayesian optimization and quantum theory-based marine predator algorithm. *Frontiers in Oncology, 14*.

[34] Rasheed, Z., Ma, Y.-K., Ullah, I., Al-Khasawneh, M., Almutairi, S. S., & Abohashrh, M. (2024). Integrating convolutional neural networks with attention mechanisms for magnetic resonance imaging-based classification of brain tumors. *Bioengineering, 11*(7), Article 7.

[35] Brain Tumor MRI Dataset. (2024). Retrieved June 15, 2024, from https://www.kaggle.com/datasets/masoudnickparvar/brain-tumor-mri-dataset.