KSÜ Mühendislik Bilimleri Dergisi, 27(3), 2024
Araştırma Makalesi

KSU J Eng Sci, 27(3), 2024
Research Article

Kahramanmaras Sutcu Imam University
Journal of Engineering Sciences

Geliş Tarihi : 06.03.2024
Kabul Tarihi : 14.06.2024

Received Date : 06.03.2024
Accepted Date : 14.06.2024

COMPUTER-AIDED DETECTION OF BRAIN TUMORS USING IMAGE PROCESSING TECHNIQUES

GÖRÜNTÜ İŞLEME TEKNIKLERI KULLANARAK BEYIN TÜMÖRLERININ BILGISAYAR DESTEKLI TESPITI

Hilal GÜVEN [1] (ORCID: 0000-0002-7461-4510)
Ahmet SAYGILI [1*] (ORCID: 0000-0001-8625-4842)

[1] Tekirdağ Namık Kemal University, Faculty of Çorlu Engineering, Department of Computer Engineering, Tekirdağ, Türkiye

*Sorumlu Yazar / Corresponding Author: Ahmet SAYGILI, asaygili@nku.edu.tr

ABSTRACT

Brain tumors are masses formed by the uncontrolled proliferation of cells in the brain. Brain tumors can be malignant or benign and can be fatal if not accurately identified at an early stage. Computer vision processing is used for early diagnosis, monitoring treatment response, and tumor classification. This study aims to detect brain tumors, a significant disease of our time, using image processing techniques. Preprocessing and data augmentation techniques were applied to a dataset of 253 images. Initially, CNNs were used for tumor detection, but transfer learning was employed for better results. Pre-trained VGG-16, DenseNet-121, ResNet-50, and MobileNet_V2 architectures were used. The model, adapted with transfer learning, achieved better performance with less data by adding a customized output layer for brain tumor detection. Experiments showed the best results with VGG-16, achieving 84.61% accuracy before data augmentation and 92.31% after augmentation. Compared to other studies, the post-augmentation accuracy rate was observed to be better than many others. The study also compares results from other deep learning architectures. Summarizing the current technological advancements in various tumor categories may help researchers understand future trends.

Keywords: Brain tumor, computer-aided detection, VGG-16, DenseNet-121, ResNet-50, MobileNet_V2

ÖZET

Beyin tümörleri, beyindeki hücrelerin kontrolsüz çoğalmasıyla oluşan kitlelerdir. Beyin tümörleri malign veya benign olabilirler ve erken aşamada doğru bir şekilde tanımlanmazsa ölümcül olabilirler. Bilgisayarlı görü işleme, erken teşhis, tedavi yanıtının izlenmesi ve tümör sınıflandırması için kullanılır. Çalışmada görüntü işleme teknikleri kullanılarak günümüzün önemli bir hastalık olan beyin tümörlerini tespit etmek amaçlanmaktadır. Bu doğrultuda, 253 görüntüden oluşan veri kümesi üzerinde önişleme teknikleri ve veri artırma teknikleri uygulanmıştır. Beyin tümörlerinin tespiti için öncelikle CNN kullanılmıştır ancak daha iyi sonuçlar elde etmek için transfer öğrenme yöntemi kullanılmıştır. Beyin tümörlerinin tespiti için önceden eğitilmiş olan VGG-16, DenseNet-121, ResNet-50, MobileNet_V2 mimarileri kullanılmıştır. Transfer öğrenme ile model, beyin tümörü tespiti için özelleştirilmiş bir çıkış katmanı ekleyerek daha az veri ile daha iyi performans elde edilmiştir. Deneyler sonucunda en iyi oranları VGG-16 mimarisi ile veri artırma öncesi %84.61, veri artırma sonrasında %92.31 doğruluk oranı elde edilmiştir. Diğer çalışmalarla karşılaştırıldığında, veri artırma sonrası elde edilen doğruluk oranının birçok çalışmadan daha iyi olduğu gözlemlenmiştir. Çalışmada ayrıca diğer derin öğrenme mimarilerinden elde edilen sonuçlar karşılaştırılarak çalışmada ayrıca sunulmuştur. Ayrıca, çeşitli tümör kategorilerindeki mevcut teknolojik ilerlemelerin özetlenmesi, araştırmacıların gelecekteki eğilimleri anlamalarına yardımcı olabilir.

Anahtar Kelimeler: Beyin tümörü, bilgisayar destekli teşhis, VGG-16, DenseNet-121, ResNet-50, MobileNet_V2

ToCite: GÜVEN, H., SAYGILI, A., (2024). COMPUTER AIDED DETECTION of BRAIN TUMORS USING IMAGE PROCESSING TECHNIQUES. *Kahramanmaraş Sütçü İmam Üniversitesi Mühendislik Bilimleri Dergisi*, 27(3), 999-1018.

INTRODUCTION

In research and clinical practice of brain tumors, which is a fatal condition endangering human life, early detection of the disease is very important to understand the multifaceted clinical picture (Atallah et al., 2024). Since manual brain tumor detection is complex, time-consuming, and error-prone, high-sensitivity automated computer-assisted diagnoses are in high demand (Sadad et al., 2021). Radiologists, engineers, and clinicians leverage medical image processing to attain a comprehensive comprehension of the anatomy of individual patients or demographic cohorts (Khan et al., 2022). Brain tumors, which greatly affect human health, are divided into benign and malignant (Nikmanesh et al., 2023). Medical robotic imaging, which uses machine learning and deep learning algorithms to analyze abnormal tissues on Magnetic Resonance Images (MRI), has led to significant advances and research in the performance and analysis of a variety of complex tasks in clinical diagnosis (Madhuri et al., 2022). Classifying brain tumors through magnetic resonance imaging (MRI) poses numerous challenges. Primarily, brain tissue exhibits a complex and irregular structure, necessitating precise segmentation for valuable clinical insights during surgeries. However, medical images of brain tissue are prone to various factors, including noise, uneven grayscale, localized volumetric variations, and image artifacts. Secondly, due to the high-density structure of the brain, poor contrast and complex structure, especially at the image edges, pose difficulties in classifying brain images (Hu et al., 2021). Manual detection of brain tumors, shape in appearance, size, nucleus, etc. It is a challenging task due to differences, and by introducing an automatic system for the timely identification of brain tumors, detection can be made faster and easier (Asad et al., 2023). In the study conducted in 2021 within the US population, it is anticipated that 88,190 new cases of malignant and benign brain tumors and other central nervous system (CNS) tumors will be diagnosed. 25,690 of these cases are benign and 62,500 are malignant. On the other hand, 83,029 deaths due to malignant brain and other central nervous system tumors were recorded over the years 2014 through 2018. According to these data, there is an average mortality rate of 4.43 per 100,000 individuals annually, resulting in a total of 16,606 deaths per year (Ostrom et al., 2021).

When reviewing the literature related to brain tumor detection and classification within the study's framework, numerous relevant studies are present. When many studies in the literature for brain tumor diagnosis are examined, it is noticed that methods such as CNN, RNN, LSTM, Transfer Learning, and GAN are used and there has been a clear increase in the number of studies on the subject recently. In the study of Çakı and Aslan, carried out a study on the CNN-FL model for brain tumor diagnosis from the MRI images they proposed. The model was trained using different activation functions (ReLU, Leaky ReLU, GELU, Swish). GELU activation function achieved a maximum success rate of 98%. CNN-FL architecture shows effective results in training with little data and comparisons with different activation functions. This study demonstrates the potential of using artificial intelligence as a helpful tool in brain tumor diagnosis (Çakı & Aslan, 2023). Khan et al. developed a deep-learning model was developed using two different MRI image datasets. First, two datasets containing 3064 and 152 MRI images were used. In the first dataset, a 23-layer convolution neural network (CNN) was applied, but since the second dataset had limited data, an overfitting problem was encountered, so transfer learning was used to solve this problem, and the VGG-16 architecture was combined with the proposed 23-layer CNN architecture. Experiments show that the proposed models attain classification accuracy of up to 97.8% and 100% for the datasets used. The results obtained in the paper show that the proposed models have impressive performance and stand out compared to other studies in the relevant literature (Khan et al., 2022). Sultan et al. In the study in which a convolutional neural network (CNN)-based deep learning (DL) model was proposed to classify brain tumors, the classification using two different datasets was focused on distinguishing certain tumor types and glioma grades. The first dataset was used to classify three different tumor types: meningioma, glioma, and pituitary tumor. The other dataset is intended to differentiate between three different grades of glioma (Grade II, Grade III, and Grade IV). These two datasets contain 233 and 73 patient data, with a total of 3064 and 516 images. For these two studies using CNN, a very good performance was achieved with an accuracy of 96.13% and 98.7%, respectively (Sultan et al., 2019). Hossain et al. introduced a study proposing a method for brain tumor detection utilizing 2D Magnetic Resonance Brain Images (MRI). In this proposed approach, the Fuzzy C-Means clustering algorithm is employed, and it is combined with both traditional classifiers and convolutional neural networks. The research was conducted using a dataset encompassing diverse tumor sizes, locations, shapes, and image densities. Among the traditional classifiers, six different algorithms including support vector machine (SVM), k-nearest neighbor (KNN), Multilayer Perceptron (MLP), Logistic Regression, Naïve Bayes, and Random Forest were utilized in the study. In the model created for the convolutional neural network (CNN), the fact that the CNN achieved 97.87% accuracy is proof that it achieved a very good performance (Hossain et al., 2019). In a study conducted by Rai and Chatterjee, the LeU-Net model, known for its efficiency on medical image datasets due to its simpler structure, fewer layers, and faster processing time, was employed to detect brain tumors from MR images.

The dataset comprised 253 images of varying sizes and pixel qualities, which underwent preprocessing steps including cropping unwanted regions, resizing, data augmentation (DA), and data distribution. Four distinct experiments were carried out on both uncropped and cropped images using four different deep-learning models (Le-Net, VGG-16, U-Net, and LeU-Net). The findings revealed that the proposed LeU-Net model achieved 98% accuracy on cropped images and 94% accuracy on uncropped images, with an impressively rapid processing time of 252.36 seconds (Rai & Chatterjee, 2021). In a study by Siar and Teshnehlab, convolutional neural networks (CNN) were employed for tumor detection. The initial image processing stage utilized CNN, resulting in a classification accuracy of 98.67%. Furthermore, the precision values were determined to be 97.34% with the Radial Basis Function (RBF) classifier and 94.24% with the Decision Tree (DT) classifier (Siar & Teshnehlab, 2019). In another study by Badža and Barjaktarović, the network's generalization ability was assessed using the 10-fold cross-validation method on an augmented image database. The best performance was achieved with record-based cross-validation on the augmented dataset, yielding an accuracy rate of 96.56%. The newly developed CNN architecture demonstrated promising potential as an effective decision-support tool in medical diagnoses for radiologists, showcasing both strong generalization ability and efficient execution speed (Badža & Barjaktarović, 2020). Kumar et al. The main goal of the study conducted is to create a convolutional neural network (CNN) that can detect and classify whether a patient is at risk for a certain disease. In the proposed method, the convolution 2D layer with the 'Leaky ReLU' activation function is used as Conv2D + Leaky ReLU. This activation function was combined and evaluated by comparing the model accuracy with Conv2D + ReLU, a traditional CNN model. The proposed model achieved a verification accuracy of 78.57%, which is higher than the traditional CNN model. However, the accuracy score of both models during training was recorded as 99.20% (Kumar et al., 2021). Govindaraj and Sandhiya present a deep transfer learning-based method for automatically classifying brain MRI images as healthy and unhealthy. Images are prepared by converting them to grayscale and applying the thresholding method. Convolutional neural networks (CNNs) were used to create the model. The study shows that the developed model achieved promising results, reaching an accuracy rate of 93-95% in the training set and 85% in the validation set (Govindaraj & Sandhiya).

In the study by Sudharson et al., methods such as SVM (support vector machine), KNN (k-nearest neighbors), and convolutional neural networks (CNN) are discussed among different techniques. In the data processing step, emphasis is placed on the efficiency of CNN for processing complex image data. The hybrid CNN approach stands out by showing significantly superior performance compared to SVM and KNN on both training and test datasets. As a result of the evaluations, it was determined that the Hybrid CNN methodology provides a significant performance advantage compared to other methods with an impressive accuracy rate of 93.22% (Sudharson et al., 2022). Chanu et al. achieved high accuracy in detecting and classifying brain tumors using YOLOv3 and CNN. The proposed methodology addresses a topic that requires expertise in fields such as image processing, object detection, and neural networks. The use of YOLOv3 provides fast and effective object detection, making it possible to quickly and accurately identify tumors. CNN is used to divide tumors into two classes: Low-Grade Glioma and High-Grade Glioma. Study results show that the proposed method achieves 97% overall accuracy. Moreover, ROC curve analysis and AUC value confirm that the classification performance of the model is high. Compared with other existing methods, the proposed method appears to be superior. In conclusion, this study demonstrated that an effective model for the detection and classification of brain tumors can be developed by using powerful techniques in the field of object detection and classification (Chanu et al., 2023). Alhalim et al. In their study, they compared different ML (machine learning) and DL (deep learning) algorithms such as VGG-16, CNNs, SVM, and KNN for brain tumor classification. These algorithms were used to classify four types of brain tumors: meningitis tumor, glioma tumor, pituitary tumor, and non-tumor state. While DL achieved 99% accuracy rates for CNN and 90% accuracy rates for VGG-16, ML did not provide suitable results and SVM achieved 91% accuracy rates. Alhalim et al. conducted empirical research on a real-time dataset featuring varying tumor sizes, locations, shapes, and image densities (Alhalim et al., 2024).

To perform the classification of brain tumor magnetic resonance images obtained from the open-source dataset, the study was carried out through transfer learning of VGG-16, which is a model that was first trained and learned general visual features. In the second part, the materials and methods section, details of the dataset used, data preprocessing, and classification model are expressed. In the third part under the experimental study title, the application used in brain tumor classification and the results obtained as a result of the application are explained. In the fourth part, the discussion section, the differences between the results of other studies obtained with different methods used in brain tumor detection and the study we developed and what can be changed to improve it further are mentioned. Finally, in the results and recommendations section in the fifth part, the brain tumor detection findings acquired as part of the study and the models used were analyzed and what new studies could be done in the future were mentioned. The general aim of the study is to emphasize the importance of early diagnosis in the detection of brain tumors, current challenges, and potential future developments and was prepared to guide future studies. The study was analyzed and compared with other studies in the literature. The contributions of the study to the literature are as follows;

• This study aims to demonstrate that better performance can be achieved with less data by using a dataset enriched with data augmentation techniques. The importance of enriching the data set is emphasized by showing how much the rate obtained before data augmentation will increase with data augmentation.

• By comparing four different transfer learning architectures, it was aimed to determine the most effective and appropriate method for brain tumor detection and to objectively evaluate the performance of these methods.

• To propose a decision support system that can support medical professionals in their decisions.

MATERIAL AND METHOD

Dataset

The dataset used in the study consists of open-source brain tumor images taken from the Kaggle site (https://www.kaggle.com/datasets/navoneel/brain-mri-images-for-brain-tumor-detection/discussion). The dataset consists of a total of 253 MRI images with two classes: 155 with tumor and 98 without tumor. The data set was obtained from volunteer patients by experts. Images in JPEG format have different resolutions. Figure 1 and Figure 2 show some original MRI images of tumor and non-tumor classes. A data augmentation method was employed to augment the dataset. Data augmentation helps reduce overlearning by increasing the generalization ability of the model. Keras' ImageDataGenerator class was used for data augmentation. Image Generator; It enables new data to be generated by making various changes to the data in the dataset. Data location, proximity, clarity, etc. is carried out by changing its properties.

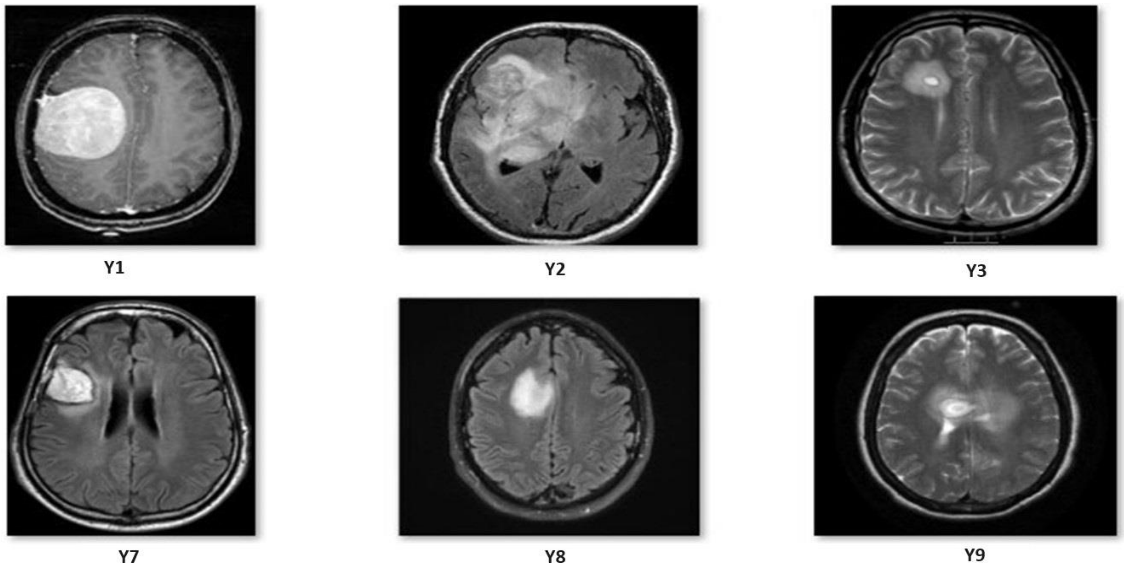


Figure 1. Example Images with Malignant Tumor

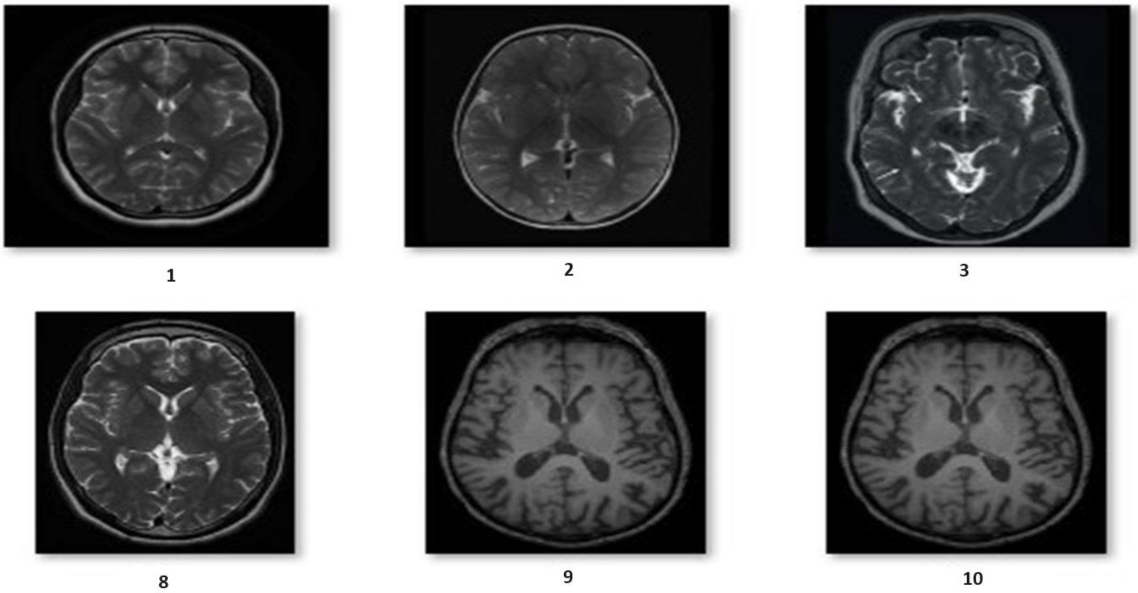


Figure 2. Example Images with Benign Tumor

Classification of Tumor

Two types of tumors shown in Figure 3 were used in the study: 1) Benign and 2) Malignant (cancer). Benign tumors are not always cancerous. Such tumors may not spread to nearby tissues or metastasize to other parts of the body. Usually, the symptoms of benign tumors are not very serious, but if it puts pressure on vital structures, for example, blood vessels or nerves, the condition can become serious (Chattopadhyay & Maitra, 2022). Benign tumors usually remain where they initially formed. While they continue to grow in these areas, they do not harm the cells around them. They do not pose a danger and are unlikely to return if such tumors are removed (Asad et al., 2023). Malignancy, derived from "mal-" meaning "bad" and "ignis" meaning "fire," refers to the region where malignant tumors, indicative of cancerous growth, are present. These tumors arise due to uncontrolled cell proliferation. If this growth persists unchecked and uninhibited, the condition may escalate into a hazardous state. Malignant tumors exhibit rapid growth and have the propensity to disseminate to other parts of the body through a process known as metastasis (Chattopadhyay & Maitra, 2022). Malignant tumors are not found in one place. They tend to constantly change their position as they grow. These cells are dangerous because they are cancerous. They invade and damage other cells nearby. Therefore, they need to be treated to be controlled (Asad et al., 2023).

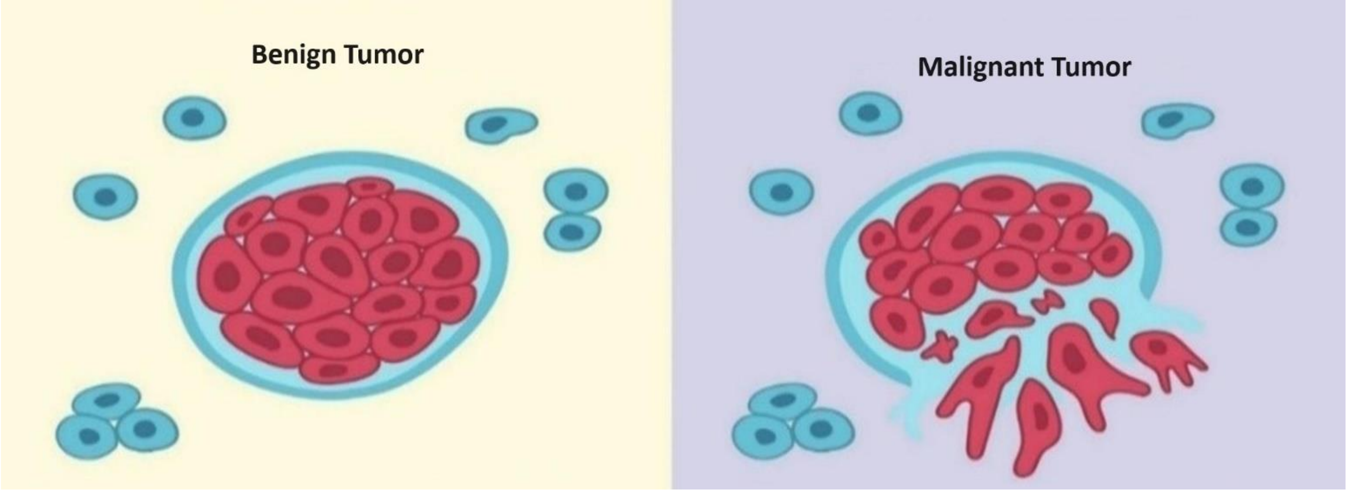


Figure 3. Benign and Malignant Tumor Structure (Asad et al., 2023)

Computer-Aided Detection System

Computer-aided diagnosis (CAD) systems aid in the diagnosis of tumors in medical images in radiological evaluations, for example, MRI images for brain tumors, tomography images for liver nodules, or mammography images for breast nodules (Chanu et al., 2023). Computer-aided diagnostics (CAD) is defined as the process of digitally creating 2D or 3D design simulations of real-world goods and products, complete with scale, precision, and physics, often in a collaborative effort, to optimize and perfect the design. Software that utilizes advanced pattern recognition and image processing techniques to assist radiologists in detecting anomalies in medical images is called Computer-Aided Diagnosis (CAD) software. It is a multidisciplinary field that blends aspects of artificial intelligence and computer vision with radiological and pathological image processing (Chan et al., 2020). Figure 4 shows the CAD working steps we used in the study.

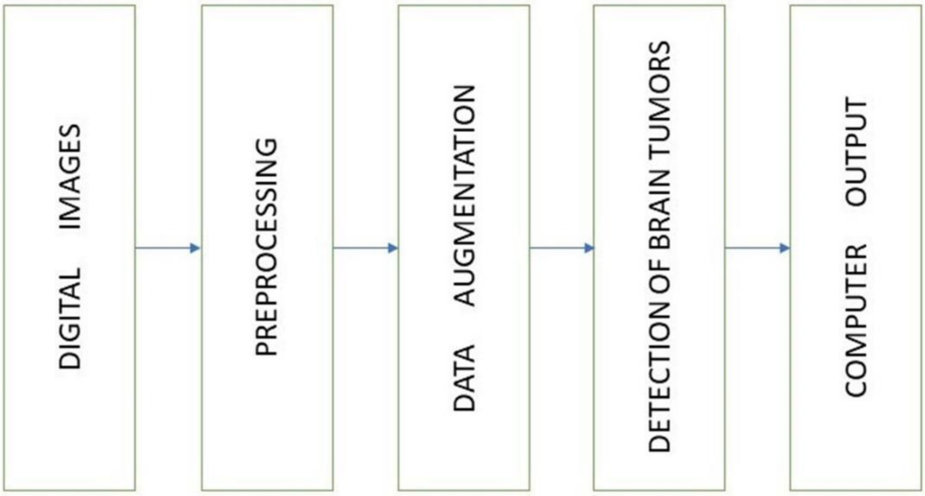


Figure 4. CAD Working Steps

Deep Learning Libraries

The capabilities of deep learning are utilized to work on big data. In our study, tensorflow and keras libraries were used. TensorFlow is a machine learning system that operates in various environments. It represents computations using data flow graphs and synchronizes them across multiple devices(Chan et al., 2020).

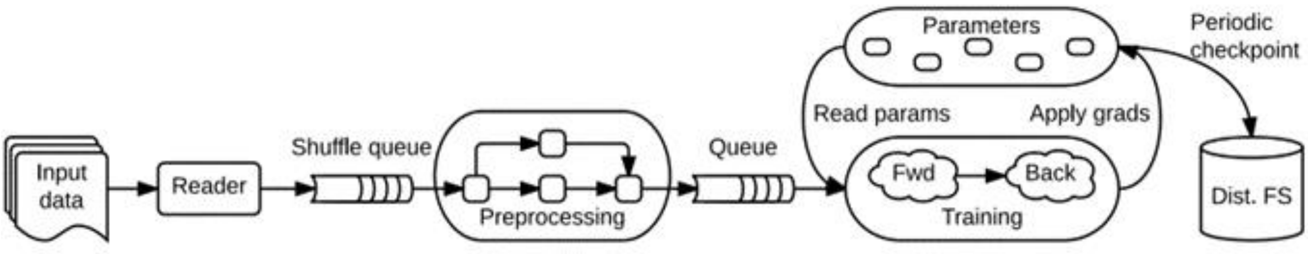


Figure 5. A Schematic Tensorflow Data Flow Chart for a Training Pipeline (Abadi et al., 2016)

Figure 5 depicts a typical training application, which operates with several subgraphs simultaneously and interacts through shared variables and queues. The basic training subgraph relies on model parameters and input batches from a queue. Several simultaneous stages of the training subgraph modify the model using varied input batches, facilitating data parallel training. Concurrent preprocessing steps are employed to fill the input queue, transforming individual input records, and a separate I/O subgraph reads records from a distributed file system. For fault tolerance, a checkpoint subgraph runs periodically. Much of TensorFlow's flexibility stems from partial and concurrent execution. The addition of mutable state and coordination via queues allows specifying a broad range of model architectures in "unprivileged" code, allowing advanced users to conduct experiments without altering the internal components of the TensorFlow runtime (Abadi et al., 2016).

Keras is a deep learning library developed in Python and handles the tasks of training and testing the deep learning model. Apart from being user-friendly, it offers high-level APIs on TensorFlow and Theano. With over 400,000 individual users as of early 2021, Keras has achieved wide acceptance in both industrial and research. The same code runs smoothly on both CPU and GPU. It allows users to rapid model prototyping and provides predefined tools for convolutional, recurrent networks, and combinations of the two (Aktürk & Serbest, 2022). Its modular structure makes it possible to easily create new models by combining neural layers, optimizers, and other components. It is supported by major companies such as Google, Microsoft, Amazon, Apple, Nvidia, and Uber. Keras offers two main models: sequential and functional API. The sequential model is simple, while the functional API allows the creation of more flexible and complex models. The functional API allows for determining the layers and then creating the model. In short, Keras is a deep learning tool that stands out with its easy-to-learn, simple model building, and wide industry support (Karabay, 2020).

Convolutional Neural Networks (CNN)

CNN is a special type of multilayer neural network most commonly used for spatial pattern analysis. Modern Convolutional Neural Networks had their beginnings in Yann LeCun and Léon Bottou's article in 1998 (Doğan & Türkoğlu, 2019). In this paper, a neural architecture called LeNet 5 is proposed to recognize handwritten digits and words with 99.2% accuracy using the MNIST dataset (LeCun et al., 1998). LeNet 5 works generally like other neural networks trained using the backpropagation algorithm, but differs in its architecture. CNNs are focused on learning spatial features, specifically the features that best describe the target class or quantity, such as edges, corners, textures, and abstract shapes. To learn these features is based on passing the input data through multiple and sequential transformations at different spatial scales (e.g., through pooling operations) via convolutions. This allows for efficient identification and integration of fundamental characteristics and advanced concepts (Arslan & Uymaz, 2022).

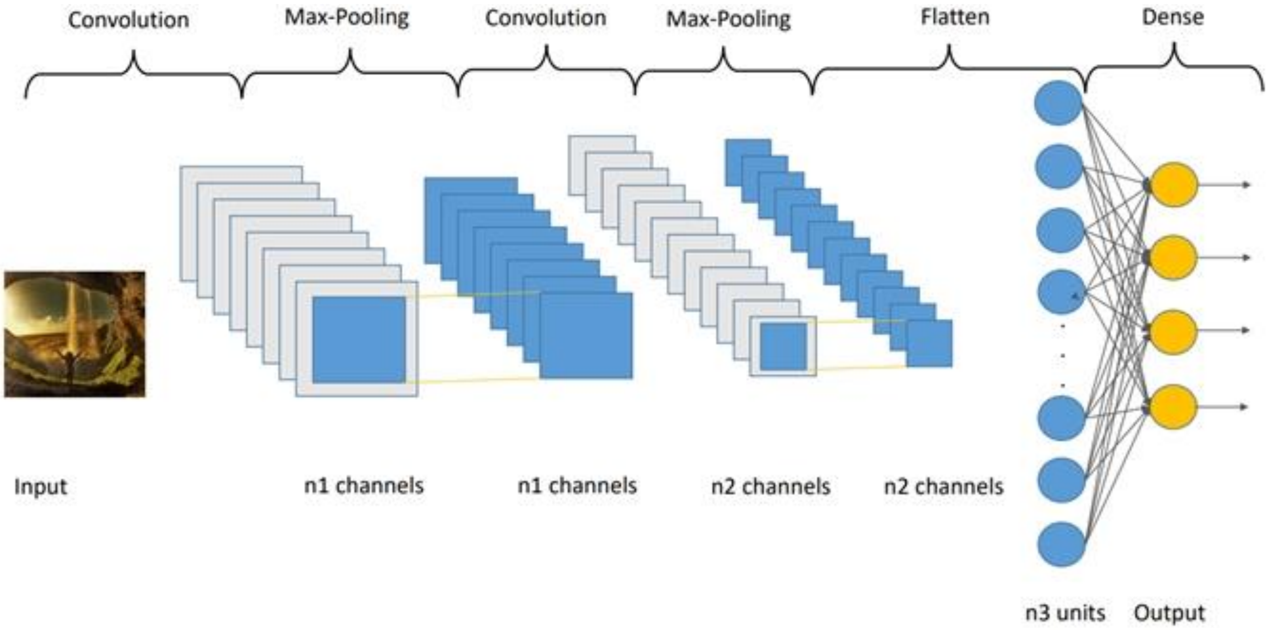


Figure 6. A Simple Convolutional Neural Network (CNN) (García-Ordás et al., 2020)

In the CNN architecture in Figure 6, various layers are shown as the Convolution Layer, ReLU Layer, Pooling Layer, Flatten, and Fully Connected Layer. The first layer, the convolutional layer, processes an image and is generally used to extract features. Images are matrices consisting of pixels with certain values. The convolution layer tries to capture certain features by traveling through a filter smaller than the original image dimensions. In this layer, the image given as input is passed through a filter. The feature map is created with the values obtained as a result of filtering. More than one filter is usually used in CNN networks, and the resulting feature maps are shaped according to the properties of the filters used. If the convolutional layer is not used, it is possible to add an input layer to the model for each pixel. However, providing the necessary links for each entry can be very difficult. Since this would require a large number of connections and processing power, it allows the operation to be performed with fewer connections by creating a feature map instead of doing this operation directly. In this way, a more effective representation can be achieved by highlighting important features (Mishra, 2020). The choice of the ReLU activation function can have an impact on the learning ability and performance of the model. Many activation functions can be used in this layer

(Doğan & Türkoğlu, 2018). The pooling layer is a layer that processes existing feature maps by dimension reduction. In this way, while processing power is decreased, extraneous features are disregarded, and attention is directed towards more significant features. Of the two different pooling techniques frequently used in CNN models, average pooling takes the average of the values in each feature map, while maximum pooling uses the maximum value in each feature map. Pooling layers are often used to minimize the size of the input before a fully connected layer. This not only uses less processing power but also helps the model focus on deeper features (Zafar et al., 2022). The flattened layer of the CNN converts the output of the convolution and pooling layers into a one-dimensional component vector for use by the dense layer of the network. This layer combines the outputs from previous layers and flattens them to create a single long component vector that will be used for final classification. This process is important to convey the features learned by the convolution and pooling layers to a dense classification layer (Dertat, 2017). The fully connected layer looks like a normal neural network connecting all neurons and usually makes up the last few layers in the network. The output from the flattened layer is fed into this fully connected layer. The output of this layer is used to classify images between different categories after training. The feature vector from the fully connected layer contains the representation that the network has learned. All inputs from this layer are connected to each activation unit of the next layer. Because the fully connected layer has a large number of parameters, it may be prone to overfitting. One of the techniques used to reduce this problem is dropout, which prevents the network from being overloaded (Vaibhav, 2018).

The architecture of the study includes layers, output sizes, and parameters for each layer. The input layer of the model, "input_1," receives images with a size of 150x150 pixels and 3 channels (RGB). This is followed by a series of convolutional (Conv2D) and maximum pooling (MaxPooling2D) layers. There is a pooling layer after each convolutional layer. Then, a global average pooling (GlobalAveragePooling2D) layer follows, with the output shape of this layer being 512. Finally, two dense (Dense) layers are added. The first dense layer consists of 256 units, while the second dense layer consists of only one unit because binary classification is performed in this model. In this research, four neural network models have been trained based on the concept of CNN architecture: ResNet50, DenseNet-121, MobileNet_V2, and VGG-16.

Resnet-50 Network Structure

In this study, ResNet-50, a variant of the popular ResNet architecture developed by Microsoft Research in 2015, is utilized. This architecture consists of 50 layers. One of its core features is the utilization of residual connections, enabling the network to learn residual functions for the input. In traditional deep neural network models, training becomes increasingly difficult as the network grows deeper, resulting in more challenging and time-consuming training processes. Hence, ResNet architecture resolves this issue through skip connections. In this approach, the output of each layer is directly added to the final layer before it (Er, 2021). ResNet-50 architecture is provided in Figure 7.

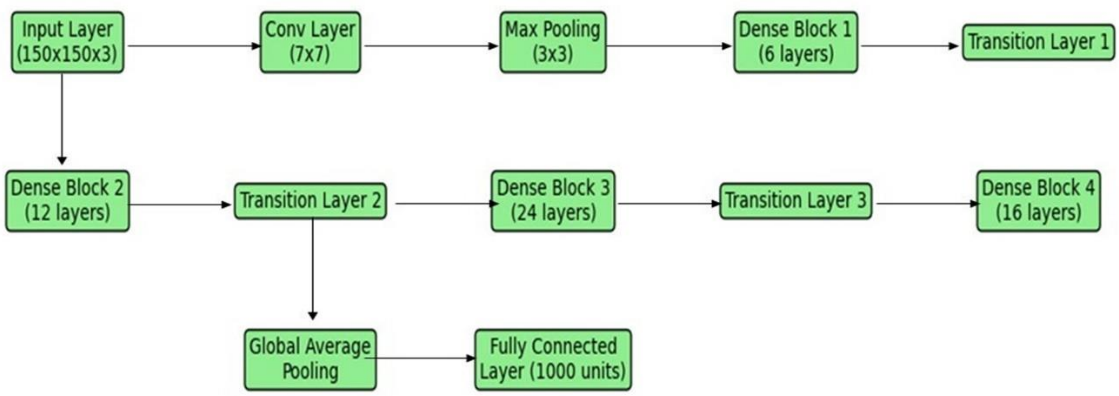


Figure 7. Resnet-50 Architecture of the Study

DenseNet-121 Network Structure

In this study, DenseNet-121 (Densely Connected Convolutional Network) connects each layer directly to all subsequent layers, creating dense connections. DenseNet121 includes a sequence of 121 interconnected convolutional layers, culminating in a fully connected layer consisting of 1000 units serving as the final output layer. In this architecture, each layer receives input from all preceding layers and passes its output to all subsequent layers. An important advantage of DenseNet architectures is their ability to facilitate feature propagation and reduce the number of parameters by enabling feature reuse. It includes 4 dense blocks containing multiple dense layers and 3 transition layers responsible for reducing the spatial dimensions of feature maps. The final classification head comprises a dense layer utilizing softmax activation for classification purposes followed by global average pooling (Dubey et al., 2022). DenseNet-121 architecture is provided in Figure 8.

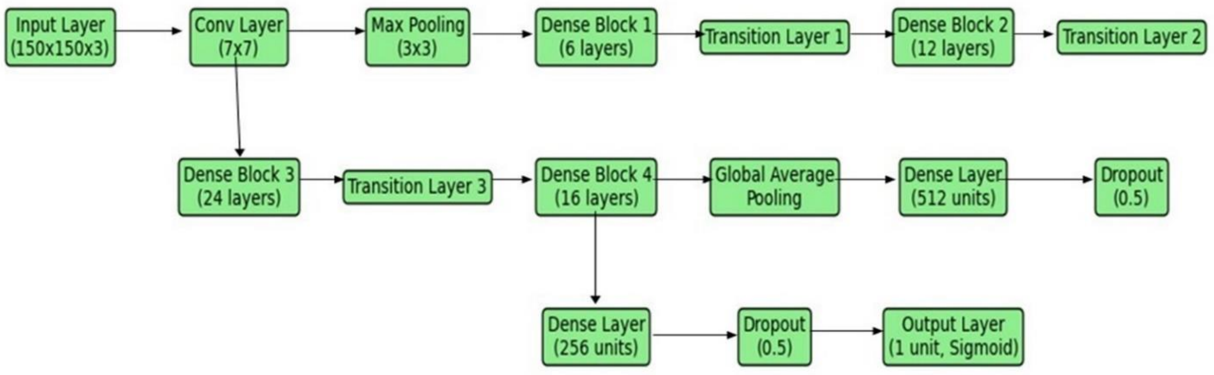


Figure 8. DenseNet-121 Architecture of the Study

MobileNet_V2 Network Structure

In this study, the MobileNet_V2 architecture is based on depthwise separable convolution. In this approach, input channels are separated into different filter channels and then combined, allowing the production of the same output as standard convolution but with fewer parameters. MobileNet_V2 consists of 28 convolutional layers with an output size of 7x7x1280 pixels. Both models accept input images of size 150x150x3 pixels (Indraswari et al., 2022). The MobileNet_V2 architecture is provided in Figure 9.

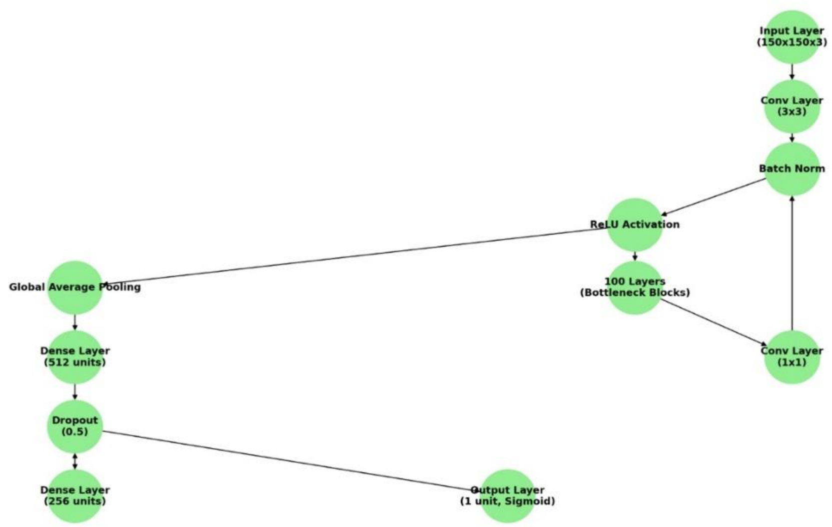


Figure 9. MobileNet_V2 Architecture of the Study

VGG-16 Network Structure

In this study, the VGG-16 deep network architecture was employed for brain tumor detection. The input layer consists of a 3-channel image input with dimensions of 150x150 pixels. The VGG-16 network comprises 13 convolution layers, each with different filter numbers and sizes. The convolution layers have a step size of 1 and are all 3x3 in size. The input layer size is 150x150x3. Max pooling layers are applied every 5 blocks, with a step size of 2 and a size of 2x2 for each pooling layer. There are a total of 2 fully connected layers, with 256 neurons in the first layer and 1 neuron in the second layer. ReLU activation is used in the first layer, while the sigmoid activation function is used in the second layer. The batch size for training the VGG-16 architecture is set to 32, and parameters are optimized using the Adam optimization method. In total, there are 19 layers, including 13 convolution layers, 5 pooling layers, 1 global average pooling layer, 2 fully connected layers, and 1 dropout layer. The VGG-16 architecture is provided in Figure 10. A transfer learning approach was utilized in this study, where convolutional weights are frozen during training to leverage pre-trained weights of the VGG-16 architecture. This allows for efficient training while solving the task of brain tumor detection.

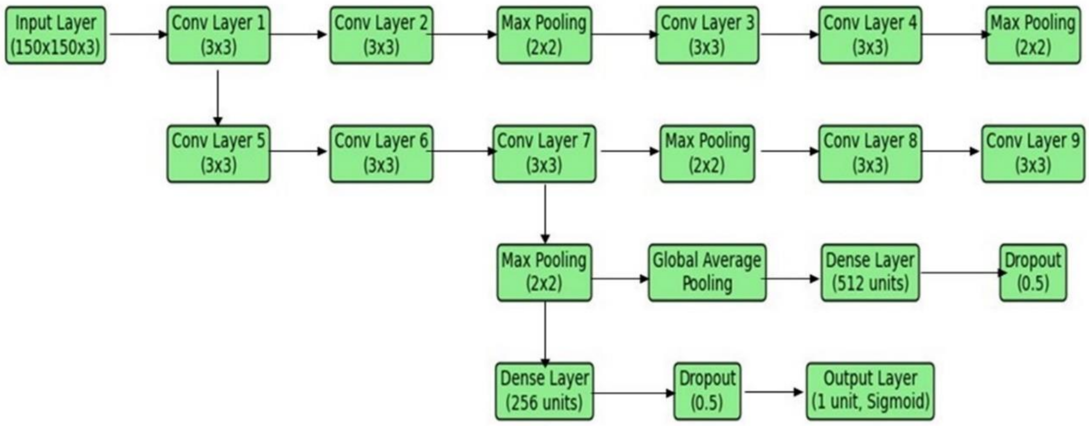


Figure 10. VGG-16 Architecture of the Study

There are some differences between these four architectures we use. One of the most obvious differences between the MobilNetV2 architecture and the VGG-16 architecture is the presence of the "Global Average Pooling" layer. While this layer does not exist in the VGG-16 architecture, it does exist in the MobilNetV2 architecture (Indraswari et al., 2022). While the MobilNetV2 architecture includes the Global Average Pooling layer compared to the ResNet-50 architecture, ResNet-50 does not have this layer. This shows that MobilNetV2 is lighter and optimized for use on portable devices, while ResNet-50 has a deeper and more complex structure. Additionally, in ResNet-50, the basic blocks are Residual Blocks. The most obvious difference between the MobilNetV2 architecture and the DenseNet-121 architecture is the Convolution layers. When we look at the convolution layers, MobilNetV2 helps reduce the computational cost and reduce the model size since there is a standard convolution layer followed by Depthwise Separable Convolution layers. In DenseNet-121, convolution layers are located in densely connected blocks. These blocks are densely interconnected and combine the outputs from previous layers (Dubey et al., 2022).

EXPERIMENTAL RESULTS

Preprocessing aims to improve the quality of MR images and convert them into a suitable form for further processing by the computer vision system. Moreover, pre-processing also helps to improve the visual aspect of MR images by enhancing the images, increasing the signal-to-noise ratio, cropping some unnecessary parts from the background, making the images smoother, and preserving edges (Sarkar et al., 2020). As a result of pre-processing in the dataset we used, a new image was obtained by cropping unnecessary parts from the background, as seen in Figure 11. Then, the data set was diversified by applying transformations such as rotating, shifting, zooming, and horizontally flipping the images through data augmentation processing.

Data augmentation was used to make the model more generalizing and to reduce the risk of overfitting. In this study, Keras' ImageDataGenerator class was used for data augmentation. The flow() method is used in the ImageDataGenerator class to read images from a large numeric array and folders containing the images. This method creates a predefined data flow for data augmentation. The ImageDataGenerator class creates new instances by performing various transformations and changes on existing images. While this method increases the size of the available dataset, it enables the model to learn more general features and perform better. In this study, the techniques used for data augmentation using the ImageDataGenerator class are as follows; The color channels of the original image were shifted by a maximum intensity of 20.0. Rotation is rotating the image by a certain angle. In the study, each image was rotated at an angle between -20 and +20 degrees by giving the value "20". Horizontal and vertical scrolling is shifting the image horizontally and vertically by a certain distance. During the data augmentation process, the images were randomly inverted horizontally. Horizontal and vertical mirroring is the process of flipping the image horizontally and vertically. Since the horizontal and vertical values were set to 0.1, each image was randomly offset by 10% in width and height. Zooming is the process of zooming in or out of the image. Since it was set to 0.1, each image was randomly zoomed in or out by 10% of its original size. After these transformations, the fill mode parameter was used to fill the empty pixels in the original images. Shear_range was set to 0.1, and a random shear angle up to 0.1 radians was applied to the image, making the model robust to such geometric distortions. The brightness_range was set to [0.9, 1.1], and the image brightness was randomly adjusted between 0.9 and 1.1 to make the model robust to different lighting conditions. By setting vertical_flip to false, the image is not randomly flipped along the vertical axis because vertical flipping is not logical for some datasets (for example, it does not make sense for images of people to be upside down). With the "Nearest" value, empty pixels are filled according to the nearest neighbor pixel value. These transformations were applied randomly to each image. The dataset is thus expanded by creating new samples. In this way, it is aimed for the model to learn more general features and perform better. These processes increase the generalization ability of the model by enabling it to better cope with images at different scales. The original Brain Tumor Mr image and the images with data augmentation techniques applied are provided in Figure 12.

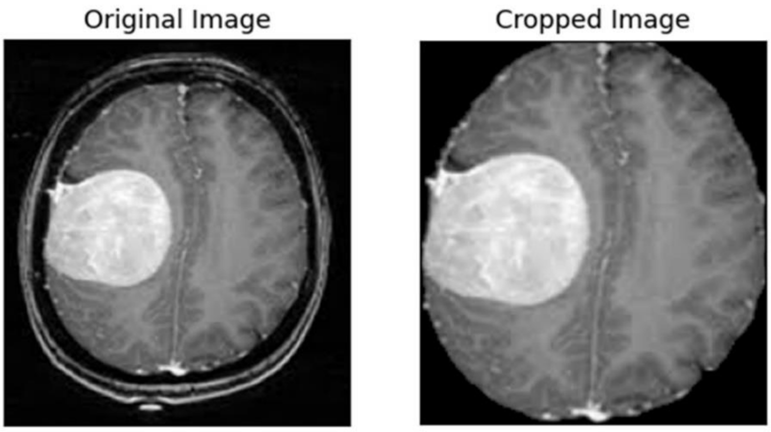


Figure 11. An Image from the Preprocessed Dataset

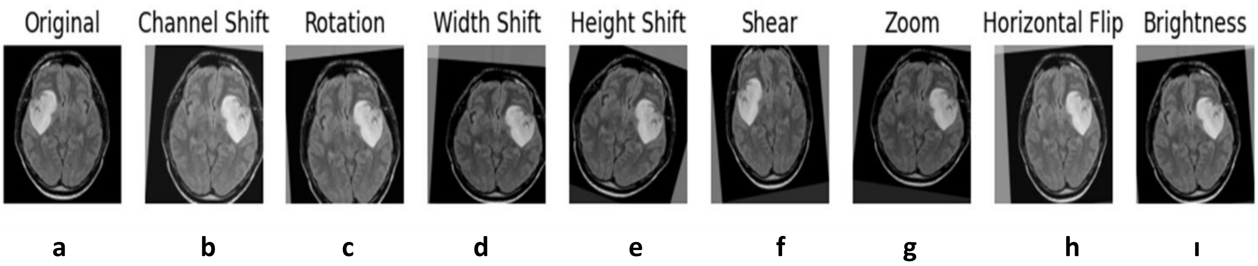
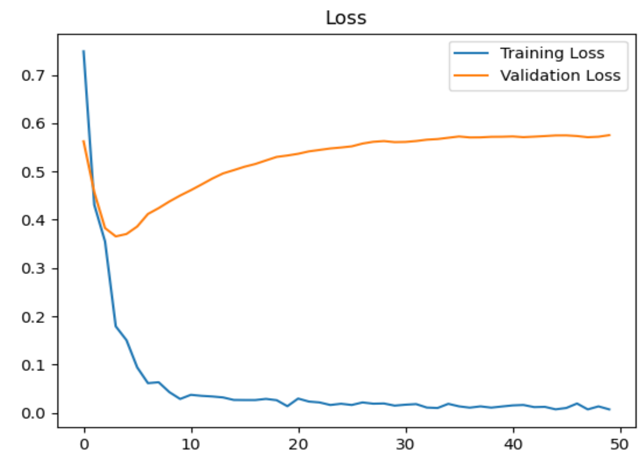


Figure 12. a. Original Brain Tumor Image **b.** Original Brain Tumor After Channel Shift **c.** Original Brain Tumor After Rotation **d.** Original Brain Tumor Image After Width Shift **e.** Original Brain Tumor Image After Heigth Shift **f.** Original Brain Tumor Image After Shear **g.** Original Brain Tumor Image After Zoom **h.** Original Brain Tumor Image After Horizontal Flip **ı.** Original Brain Tumor Image After Brightness

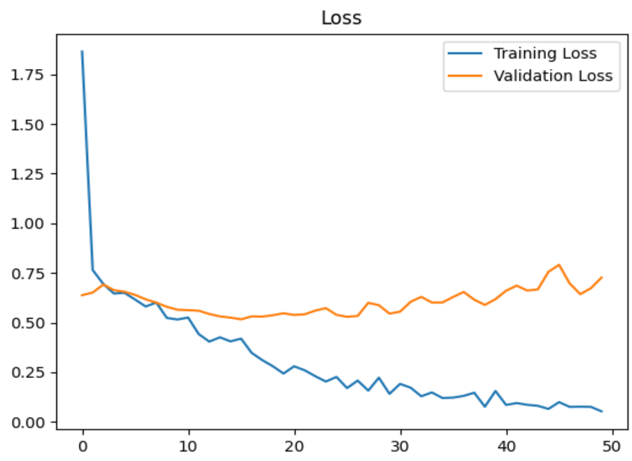
Table 1 shows the partitioning of the dataset and the dimensions of each partition. There are 253 samples in total. This study requires separating the image data for the learning, validation, and testing phases. Data augmentation was applied only on the training data because using it on the test and validation datasets would cause overfitting. As a result of the data augmentation process, the number of training samples was increased from 177 to 9600. Each sample is represented as a 150x150 pixel image with 3 channels (i.e. RGB).

Table 1. Dataset Information After Data Augmentation

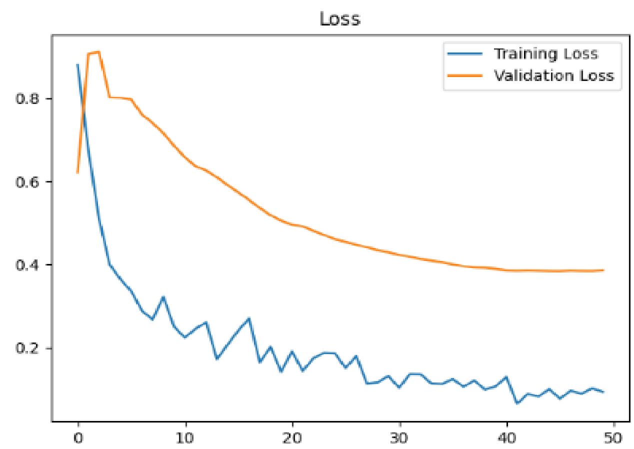
Dataset Partitioning	Number of Samples	Sample Dimensions	Label Dimensions
Total	253	-	-
Training	9600	150x150x3	9600x1
Validation	39	150x150x3	39x1
Test	38	150x150x3	38x1



a



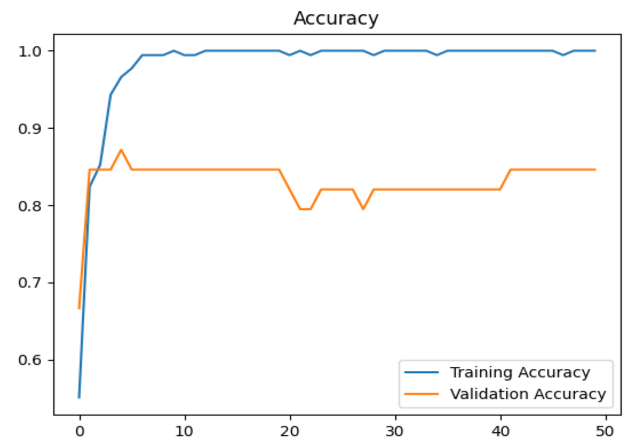
b



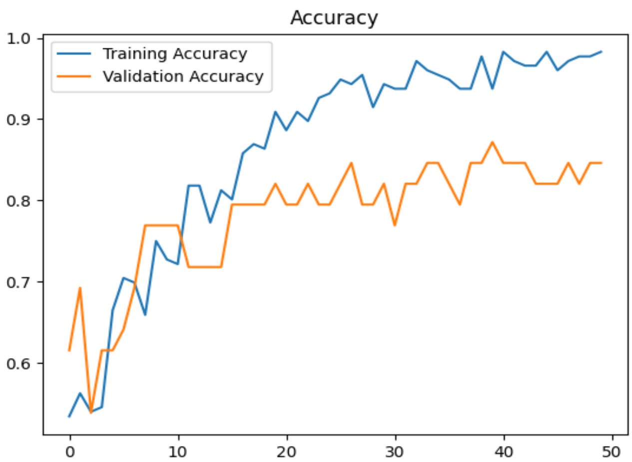
c

d

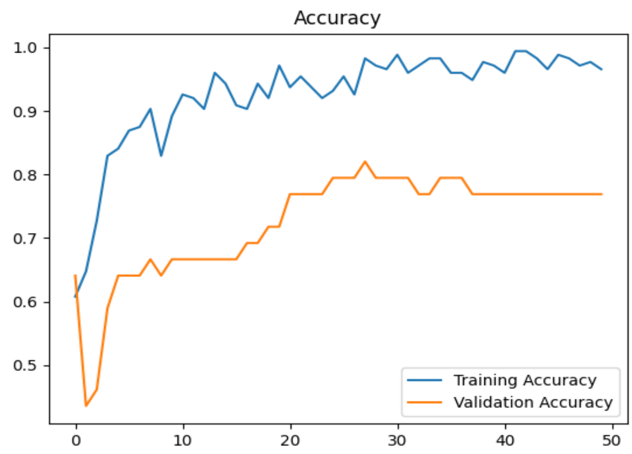
Figure 13. a. ResNet-50 Performance Loss Status Chart Representation Without Data Augmentation **b.** VGG-16 Performance Loss Status Chart Representation Without Data Augmentation **c.** DenseNet-121 Performance Loss Status Chart Representation Without Data Augmentation **d.** MobileNet_V2 Performance Loss Status Chart Representation Without Data Augmentation



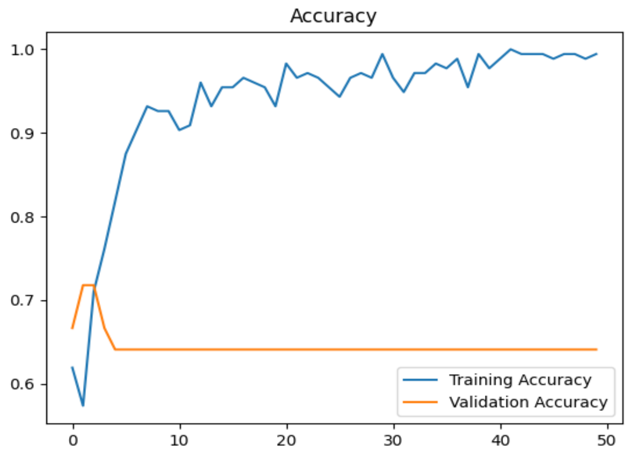
a



b



c



d

Figure 14. a. ResNet-50 Performance Success Status Chart Representation Without Data Augmentation **b.** VGG-16 Performance Success Status Chart Representation Without Data Augmentation **c.** DeseNet121 Performance Success Status Chart Representation Without Data Augmentation **d.** MobileNet_V2 Performance Success Status Chart Representat

a

True Label	Predicted (-)	Predicted (+)
Actual (-)	12	3
Actual (+)	3	21

Predicted Label

b

True Label	Predicted (-)	Predicted (+)
Actual (-)	10	5
Actual (+)	1	23

Predicted Label

c

True Label	Predicted (-)	Predicted (+)
Actual (-)	10	5
Actual (+)	4	20

Predicted Label

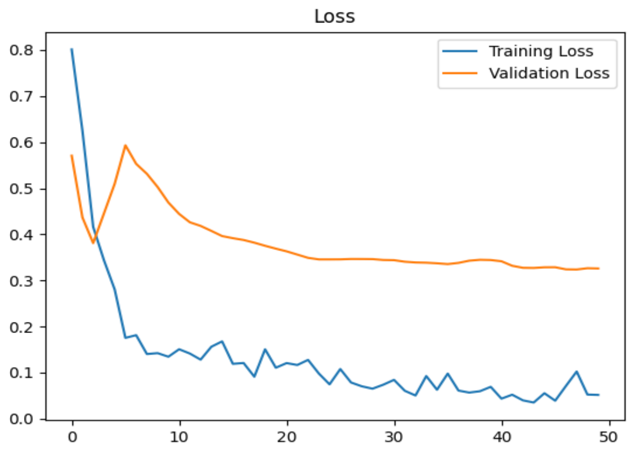
d

True Label	Predicted (-)	Predicted (+)
Actual (-)	1	14
Actual (+)	0	24

Predicted Label

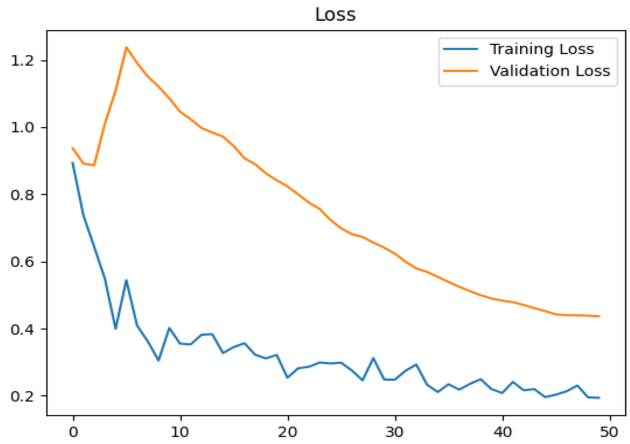
Figure 15. a. ResNet-50 Confusion Matrix Without Data Augmentation **b.** VGG-16 Confusion Matrix Without Data Augmentation **c.** DenseNet-121 Confusion Matrix Without Data Augmentation **d.** MobileNet_V2 Confusion Matrix Without Data Augmentationion Without Data Augmentation

The Resnet-50, VGG-16, DenseNet-121, and MobileNet_V2 architectures have been implemented for brain tumor detection using transfer learning with TensorFlow and Keras libraries. The dataset consists of a total of 253 MRI images, with 155 being tumor and 98 tumor-free, forming two classes. Results obtained before data augmentation are shown in Figures 13, 14, and 15. Figure 13 illustrates the corresponding loss value, Figure 14 shows training and validation accuracy over epochs, and Figure 15 displays the Confusion Matrix. In this study, MobileNet_V2, DenseNet-121, Resnet-50 and VGG-16 achieved accuracy rates of 64.10%, 76.92%, 84.61%, and 84.61%, respectively.

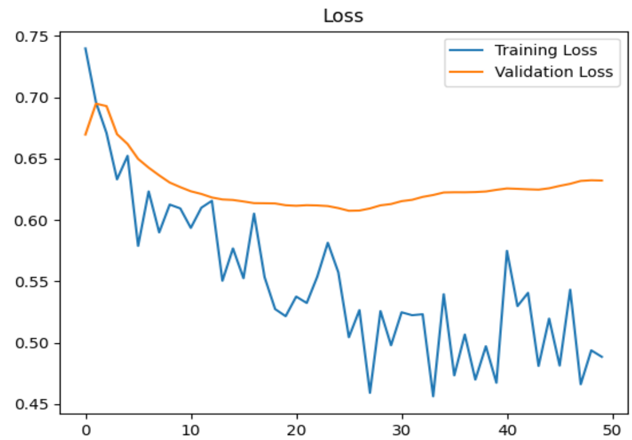


a

b

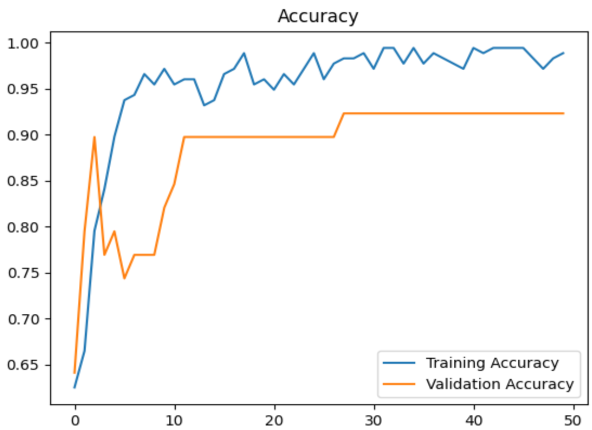


c

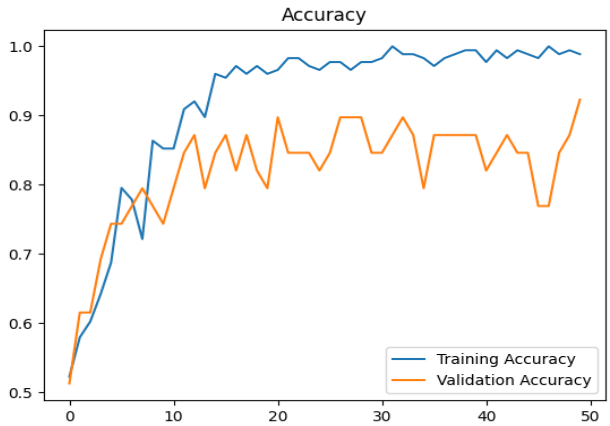


d

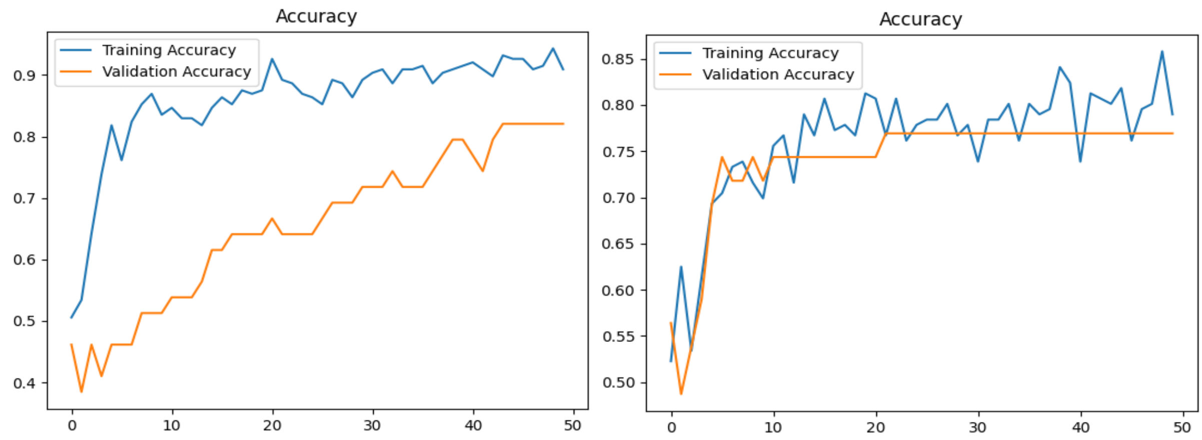
Figure 16. a. ResNet-50 Data Augmented Performance Loss Status Graph Representation **b.** VGG-16 Data Augmented Performance Loss Status Graph Representation **c.** DeseNet121 Data Augmented Performance Loss Status Graph Representation **d.** MobileNet_V2 Data Augmented Performance Loss Status Graph Representation



a



b



c **d**

Figure 17. a. ResNet-50 Data Augmented Performance Success Status Chart Representation **b.** VGG-16 Data Augmented Performance Success Status Chart Representation **c.** DeseNet121 Data Augmented Performance Success Status Chart Representation **d.** MobileNet_V2 Data Augmented Performance Success Status Chart Representation

a

True Label	Predicted (-)	Predicted (+)
Actual (-)	13	2
Actual (+)	1	23

Predicted Label

b

True Label	Predicted (-)	Predicted (+)
Actual (-)	13	2
Actual (+)	1	23

Predicted Label

c

True Label	Predicted (-)	Predicted (+)
Actual (-)	13	2
Actual (+)	5	19

Predicted Label

d

True Label	Predicted (-)	Predicted (+)
Actual (-)	6	9
Actual (+)	0	24

Predicted Label

Figure 18. a. ResNet-50 Confusion Matrix with Data Augmentation **b.** VGG-16 Confusion Matrix with Data Augmentation **c.** DenseNet-121 Confusion Matrix with Data Augmentation **d.** MobileNet_V2 Confusion Matrix with Data Augmentation

In the subsequent process, data augmentation was performed using ImageDataGenerator to augment the dataset, and the results obtained are shown in Figures 16, 17, and 18. Figure 16 illustrates the corresponding loss value, Figure 17 shows training and validation accuracy over epochs, and Figure 18 displays the Confusion Matrix. In this study, MobileNet_V2, DenseNet-121, Resnet-50, and VGG-16 achieved accuracy rates of 76.92%, 82.02%, 92.30%, and 92.31% respectively.

Performance evaluation criteria classification models generally measure their performance based on the relationships between the classes predicted by the classifier and the actual classes to evaluate their success. Essentially, these evaluation criteria aim to measure the overall performance of the classification model by evaluating the model's accuracy, precision, sensitivity, and F1 score. Equations for performance evaluation criteria are given in 1-4.

$$\text{Accuracy: } \frac{TP+TN}{TP+TN+FP+FN} \times 100 \quad (1)$$

Precision: $\frac{\text{TP}}{\text{TP+FP}}$ x 100 (2)

Recall: $\frac{\text{TP}}{\text{TP+FN}}$ x 100 (3)

F1-score: 2 x $\frac{\text{Precision}*\text{Recall}}{\text{Precision+Recall}}$ (4)

Figure 15 depicts the confusion matrix of the dataset without data augmentation. Figure 18, on the other hand, displays the confusion matrix. The classification results obtained using the confusion matrix are presented in Table 2.

Table 2. Classification Results

Model	Precision	Recall	F1-score	Accuracy (%)
Resnet-50(No Data augmentation)	87.50	87.50	87.50	84.61
VGG-16 (No Data augmentation)	82.21	95.83	88.46	84.61
DenseNet-121(No Data augmentation)	80.00	83.33	81.63	76.92
MobilNetV2(No Data augmentation)	63.15	100.00	77.41	64.10
Resnet-50(With Data augmentation)	92.00	95.83	93.87	92.30
VGG-16 (With Data augmentation)	92.00	95.83	93.87	92.31
DenseNet-121(With Data augmentation)	90.47	79.16	84.44	82.05
MobilNetV2(With Data augmentation)	72.72	100.00	84.21	76.92

This study aimed to demonstrate the most effective architecture for brain tumor detection by using various architectures. When studies conducted using the same dataset as ours were examined, it was shown in Table 3 that very successful results have been achieved in recent years with various methods in brain tumor detection. When studies on brain tumors were examined, it was determined that various classifiers and different activation functions were used in addition to the CNN deep learning model. Accordingly, the success rates of the models in accurately predicting brain tumors are listed. When Table 3 is examined, it can be seen that successful deep-learning studies have been conducted for the detection of brain tumors. In experiments, it was observed that the accuracy rates obtained from different deep learning architectures and segmentation models varied between 55.00% and 95.00%. When comparing the proposed methods, among the architectures without data augmentation, Resnet-50 and VGG-16 achieved the highest performance with an accuracy rate of 84.61%. When examining the architectures with data augmentation, VGG-16 achieved the highest performance with an accuracy rate of 92.31%. It was observed that data augmentation resulted in better performance in all architectures. The VGG-16 model we used has achieved higher results compared to the studies we compared. In the study conducted by Saxena et al., the VGG-16 model achieved an accuracy rate of 90.00%, while in our study, we have reached the highest model rate in the VGG-16 model with an accuracy rate of 92.31%. Although the rate we obtained is good, better results can be achieved with some changes. Changes that can be made include expanding and diversifying the dataset to improve the model's generalization ability. Adding more tumor and tumor-free samples can help the model learn better. Additionally, working with images of different resolutions can help the model adapt to a wider range of data. Furthermore, more testing and validation studies are needed before such models can be used in clinical practice. Hyperparameter adjustments can be made to improve the model's performance. Setting parameters such as learning rate to optimal values can enable the model to learn better. Additionally, solving the class imbalance problem by using weighted loss functions can be considered. By conducting an error analysis study on misclassified examples in the model, the weak points in the model can be determined, understanding why the model makes these errors, and improvements can be made in this regard. Particularly, examining false positive and false negative situations will contribute more to us. Finally, different approaches used in similar studies in the literature should be tried to determine which one provides the better result.

Table 3. Performance Comparisons of Some Methods

Studies with the same dataset	Methods	Accuracy (%)
(Saxen et al., 2020) (With Data augmentation)	Inception-V3, VGG-16, Resnet-50	Sequentially 55.00, 90.00, 95.00
(Zailan et al., 2022) (No Data augmentation)	VGG-16, Inception V3, MobileNet_V2	Sequentially 78.95, 84.21, 85.53
(Macdonald & Engelhardt, 2010)	VGG-16, ResNet-50, Inception V3	Sequentially 90.00, 95.00, 55.00
This Study (No Data augmentation)	MobilNetV2, DenseNet-121, Resnet-50, VGG-16	Sequentially 64.10, 76.92, 84.61, 84.61
This Study (With Data augmentation)	MobilNetV2, DenseNet-121, Resnet-50, VGG-16	Sequentially 76.92, 82.05, 92.30, 92.31

CONCLUSION

The dataset used in this study was obtained from the Kaggle website. The dataset consists of a total of 253 MRI images, divided into two classes: 155 with tumors and 98 without tumors. The dataset was obtained from voluntary patients by experts. The images in JPEG format have different resolutions. Since the number of training data in the dataset is low, data augmentation was applied. As a result of the data augmentation process, 9600 training data were obtained. To demonstrate the impact of data augmentation, the study was conducted on both the original dataset and the augmented dataset. Four different models were used: VGG-16, DenseNet-121, ResNet-50, and MobileNet_V2. When examining the experimental results, among the architectures used without data augmentation, ResNet-50 and VGG-16 achieved the highest performance with an accuracy rate of 84.61%, while among the architectures used with data augmentation, VGG-16 achieved the highest performance with a classification accuracy rate of 92.31%. This indicates the extent to which the performance of the data augmentation process can be enhanced by increasing the diversity of the training dataset. When examining the test loss values, the best result for the VGG-16 architecture without data augmentation was 0.72, and with data augmentation, it was 0.36. This low loss value indicates that the model can effectively identify the patterns seen in the training data in the test data and generalize successfully. Class imbalance can be addressed by using weighted loss functions or by balancing the dataset to achieve better rates. To understand why the model makes these errors, a detailed error analysis should be conducted on the examples misclassified by the model, especially focusing on false positive and false negative cases. To improve the model's performance, hyperparameter adjustments (e.g., learning rate, dropout rates) can be made. This can help the model achieve better generalization ability. Parameter and optimization changes should be made, and results should be compared. In this way, the best result can be found. Additionally, collecting more data for the dataset can achieve better rates. The lack of data in the dataset used in this study can be considered a point for improvement. To achieve better results in the study, we might consider increasing the actual number of images in the dataset used. To make the study more useful, in future studies, other open-source brain tumor datasets in the literature can be used to develop more deep learning-based models to assist physicians in detecting brain tumors. However, in addition to classification problems, abnormal images (disease, positive, tumor) may need to be manually segmented by physicians. In this context, AI-supported automatic segmentation studies can be conducted. These methods can provide more support to physicians by improving the diagnostic process and playing a more effective role in treatment planning.

This study demonstrates that despite initially having limited data in our training dataset, various data augmentation techniques have been employed to enrich the dataset. This technically proves that even with a small dataset, good results can be achieved through the utilization of data augmentation techniques. Four different transfer learning architectures have been compared, and upon reviewing other conducted studies, it is noticed that researchers have reached varying results among the models. Overall, in our study and comparison, to others, the VGG-16 model has achieved results closely aligned with each other. Our study demonstrates the efficacy of image processing in providing promising results, thereby substantiating its potential as a supportive decision-making tool for medical professionals.

REFERENCES

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G. & Isard, M. (2016). TensorFlow: a system for Large-Scale machine learning. In *12th USENIX symposium on operating systems design and implementation* (OSDI 16) (pp. 265-283).

Aktürk, S. & Serbest, K. (2022). Nesne Tespiti İçin Derin Öğrenme Kütüphanelerinin İncelenmesi. *Journalof Smart Systems Research*, 3(2), 97-119

Alhalim, A., Abd Alrahman, G., Hussain Hassan, N. M. & Nashat, A. A. (2024). Computer-Aided Diagnosis And Detection For Brain Cancer. *Fayoum University Journal of Engineering*, *7*(1), 49-62. https://doi.org/10.21608/FUJE.2023.221477.1052

Arslan, Ö. & Uymaz, S. A. (2022). Classification of Invoice Images By Using Convolutional Neural Networks. *Journal of Advanced Research in Natural and Applied Sciences*, *8*(1), 8-25. https://doi.org/10.28979/jarnas.953634

Asad, R., Rehman, S. U., Imran, A., Li, J., Almuhaimeed, A. & Alzahrani, A. (2023). Computer-Aided Early Melanoma Brain-Tumor Detection Using Deep-Learning Approach. *Biomedicines*, *11*(1), 184. https://doi.org/10.3390/biomedicines11010184

Atallah, O., Badary, A., Almealawy, Y. F., Sanker, V., Awuah, W. A., Abdul-Rahman, T., Alrubaye, S. N. & Chaurasia, B. (2024). Non-colloid-cyst primary brain tumors: A systematic review of unexpected fatality. *Journal of Clinical Neuroscience*, *119*, 129-140. https://doi.org/10.1016/j.jocn.2023.11.022

Badža, M. M. & Barjaktarović, M. Č. (2020). Classification of brain tumors from MRI images using a convolutional neural network. *Applied Sciences*, *10*(6), 1999. https://doi.org/10.3390/app10061999

Chan, H. P., Hadjiiski, L. M. & Samala, R. K. (2020). Computer-aided diagnosis in the era of deep learning. *Medical physics*, *47*(5), e218-e227. https://doi.org/10.1002/mp.13764

Chanu, M. M., Singh, N. H., Muppala, C., Prabu, R. T., Singh, N. P. & Thongam, K. (2023). Computer-aided detection and classification of brain tumor using YOLOv3 and deep learning. *Soft Computing*, *27*(14), 9927-9940. https://doi.org/10.1007/s00500-023-08343-1

Chattopadhyay, A. & Maitra, M. (2022). MRI-based brain tumour image detection using CNN based deep learning method. *Neuroscience informatics*, *2*(4), 100060. https://doi.org/10.1016/j.neuri.2022.100060

Aslan, T. & Çakı, E. E. (2023). Beyin tümörü teşhisinde CNN-FL modeli ağ performansının aktivasyon fonksiyonlarına göre karşılaştırılması. *Journal of Scientific Reports-B,* 008, 43-54.

Dertat, A. (2017). Applied deep learning-part 4: Convolutional neural networks. Towards Data Science, 26. https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2 Accessed 14.02.2024.

Doğan, F. & Türkoğlu, (2019). Derin öğrenme modelleri ve uygulama alanlarına ilişkin bir derleme. *Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi*, 10(2), 409-445. https://doi.org/10.24012/dumf.411130.

Doğan, F. & Türkoğlu, İ. (2018). "Derin Öğrenme Algoritmalarının Yaprak Sınıflandırma Başarımlarının Karşılaştırılması." *Sakarya University Journal of Computer and Information Sciences*, 1(1), 10-21.

Dubey, N., Bhagat, E., Rana, S., & Pathak, K. (2022). A novel approach to detect plant disease using DenseNet-121 neural network. In *Smart Trends in Computing and Communications: Proceedings of SmartCom 2022* (pp. 63-74). Singapore: Springer Nature Singapore. https://doi.org/10.1007/978-981-16-9967-2_7

Er, M. B. (2021). Önceden Eğitilmiş Derin Ağlar İle Göğüs Röntgeni Görüntüleri Kullanarak Pnömoni Siniflandirilmasi. *Konya Journal of Engineering Sciences*, *9*(1), 193-204. https://doi.org/10.36306/konjes.794505

García-Ordás, M. T., Benítez-Andrades, J. A., García-Rodríguez, I., Benavides, C. and Alaiz-Moretón, H. (2020). Detecting respiratory pathologies using convolutional neural networks and variational autoencoders for unbalancing data. *Sensors*, *20*(4), 1214. https://doi.org/10.3390/s20041214

Govindaraj, S. and Sandhiya, G. *Brain Tumor Detection Using Convolutional Neural Network With Image Processing*. *2019 1st international conference on advances in science, engineering and robotics technology (ICASERT)*. IEEE. https://www.doi.org/10.56726/irjmets38709

Hossain, T., Shishir, F. S., Ashraf, M., Al Nasim, M. A. and Shah, F. M. (2019). Brain tumor detection using convolutional neural network. 2019 1st international conference on advances in science, engineering and robotics technology (ICASERT), 1-6. IEEE.

Hu, M., Zhong, Y., Xie, S., Lv, H. and Lv, Z. (2021). Fuzzy system based medical image processing for brain disease prediction. *Frontiers in Neuroscience*, *15*, 714318. https://doi.org/10.3389/fnins.2021.714318

Indraswari, R., Rokhana, R., & Herulambang, W. (2022). Melanoma image classification based on MobileNetV2 network. *Procedia computer science*, *197*, 198-207. https://doi.org/10.1016/j.procs.2021.12.132

Karabay, A. Keras Nedir? [What is Keras?]. Retrieved from https://www.karabayyazilim.com/blog/python/keras-nedir-2020-02-08-225241 Accessed 08.02.2024

Khan, M. S. I., Rahman, A., Debnath, T., Karim, M. R., Nasir, M. K., Band, S. S., Mosavi, A. and Dehzangi, I. (2022). Accurate brain tumor detection using deep convolutional neural network. *Computational and Structural Biotechnology Journal*, *20*, 4733-4745. https://doi.org/10.1016/j.csbj.2022.08.039

Kumar, G., Kumar, P. and Kumar, D. (2021). Brain tumor detection using convolutional neural network. 2021 *IEEE International Conference on Mobile Networks and Wireless Communications (ICMNWC)* (pp. 1-6). IEEE. https://doi.org/10.1109/ICMNWC52512.2021.9688460

LeCun, Y., Bottou, L., Bengio, Y. and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278-2324. https://doi.org/10.1109/5.726791

Macdonald, D., & Engelhardt, G. (2010). 2.39 predictive modeling of corrosion. *Shreir's Corros, Elsevier BV*, *2*, 1630-1679.

Madhuri, G. S., Mahesh, T. and Vivek, V. (2022). A novel approach for automatic brain tumor detection using machine learning algorithms. In *Big data management in Sensing* (pp. 87-101). River Publishers. https://doi.org/10.1201/9781003337355-7

Mishra, M. (2020). Convolutional neural networks, explained. *Towards Data Science*, *26*. https://towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939 Accessed 14.02.24.

Nikmanesh, Y., Mohammadi, M. J., Yousefi, H., Mansourimoghadam, S., & Taherian, M. (2023). The effect of long-term exposure to toxic air pollutants on the increased risk of malignant brain tumors. *Reviews on Environmental Health*, *38*(3), 519-530. https://doi.org/10.1515/reveh-2022-0033

Ostrom, Q. T., Cioffi, G., Waite, K., Kruchko, C., & Barnholtz-Sloan, J. S. (2021). CBTRUS statistical report: primary brain and other central nervous system tumors diagnosed in the United States in 2014–2018. *Neuro-oncology*, *23*(Supplement_3), iii1-iii105. https://doi.org/10.1093/neuonc/noab200

Rai, H. M., & Chatterjee, K. (2021). 2D MRI image analysis and brain tumor detection using deep learning CNN model LeU-Net. *Multimedia Tools and Applications*, *80*, 36111-36141. https://doi.org/10.1007/s11042-021-11504-9

Sadad, T., Rehman, A., Munir, A., Saba, T., Tariq, U., Ayesha, N., & Abbasi, R. (2021). Brain tumor detection and multi-classification using advanced deep learning techniques. *Microscopy Research and Technique*, *84*(6), 1296-1308. https://doi.org/10.1002/jemt.23688

Sarkar, S., Kumar, A., Chakraborty, S., Aich, S., Sim, J.-S., & Kim, H.-C. (2020). A CNN based approach for the detection of brain tumor using MRI scans. *Test Engineering and Management*, *83*, 16580-16586.

Saxena, P., Maheshwari, A., & Maheshwari, S. (2020). Predictive modeling of brain tumor: a deep learning approach. In *Innovations in Computational Intelligence and Computer Vision: Proceedings of ICICV 2020* (pp. 275-285). Singapore: Springer Singapore. https://doi.org/10.1007/978-981-15-6067-5_30

Siar, M., & Teshnehlab, M. (2019). Brain tumor detection using deep neural network and machine learning algorithm. In *2019 9th international conference on computer and knowledge engineering* (ICCKE) (pp. 363-368). IEEE. https://doi.org/10.1109/ICCKE48569.2019.8964846

Sudharson, K., Sermakani, A., Parthipan, V., Dhinakaran, D., Petchiammal, G. E., & Usha, N. (2022). Hybrid Deep Learning Neural System for Brain Tumor Detection. In *2022 2nd International Conference on Intelligent Technologies (CONIT)* (pp. 1-6). IEEE. https://doi.org/10.1109/CONIT55038.2022.9847708

Sultan, H. H., Salem, N. M., & Al-Atabany, W. (2019). Multi-classification of brain tumor images using deep neural network. *IEEE access*, *7*, 69215-69225. 10.1109/ACCESS.2019.2919122

Vaibhav, R. Fully Connected Layer. Medium. https://medium.com/@vaibhav1403/fully-connected-layer-f13275337c7c Accessed 10.02.24

Zafar, A., Aamir, M., Mohd Nawi, N., Arshad, A., Riaz, S., Alruban, A., Dutta, A. K., & Almotairi, S. (2022). A comparison of pooling methods for convolutional neural networks. *Applied Sciences*, *12*(17), 8643. https://doi.org/10.3390/app12178643

Zailan, Z. N., Mostafa, S. A., Abdulmaged, A. I., Baharum, Z., Jaber, M. M., & Hidayat, R. (2022). Deep Learning Approach for Prediction of Brain Tumor from Small Number of MRI Images. *JOIV: International Journal on Informatics Visualization*, *6*(2-2), 581-586.