Acta Scientiarum
http://periodicos.uem.br/ojs/acta
ISSN on-line: 1807-8664
Doi: 10.4025/actascitechnol.v47i1.71395

Majority Voting-Based Tumor Detection: Brain Tumor Detection on X-Ray and MRI Images Using Fusion of Hybrid Learning Methods

Duygu Çelik Ertuğrul*, Önsen Toygar and Hasan Karataş

[1]Department of Computer Engineering, Faculty of Engineering, Eastern Mediterranean University, 99628, Famagusta, North Cyprus, Mersin 10, Turkey.
*Author for correspondence. E-mail: duygu.celik@emu.edu.tr

ABSTRACT. Brain tumor is a crucial health problem that affects people's lives and their life quality. The treatment of this disease varies according to the size, type, location, and condition of the tumor detected. Therefore, the treatment of this disease may vary from person to person. Early diagnosis of brain tumor in individuals is very important as it can change the course of treatment. The aim of this study is to provide a hybrid learning solution for early detection of brain tumors on Magnetic Resonance Imaging (MRI) and X-Ray scans. In this study, two separate datasets involving a total of 7600 MRI and X-Ray scans, available to users from Kaggle, were used. In the experimental part of this study, the 7600 MRI and X-Ray scans were trained and tested using a number of well-known learning models which are; XGBoost, Convolutional Neural Networks (CNN), ResNet50, DenseNet121 and AlexNet. After the experimental studies, the accuracy, sensitivity, specificity, precision, recall and F1-Score metrics of each of these models were obtained and compared. In addition, taking into account the results obtained from these models, a fusion approach named "Majority Voting" has been applied and the performance value of the system has been successfully increased. In summary, the performance results obtained from the used model are 98.98% for XGBoost, 99.75% for CNN, 99.65% for DenseNet121, 97.21% for ResNet50 and 99.84% for AlexNet. The accuracy after the "Majority Voting" approach applied is 100.00%. The results of the experimental studies demonstrate the promise of the proposed hybrid learning system with the Majority Voting approach and emphasizing its feasibility, effectiveness, and efficiency in processing both MRI and X-Ray tumor scanning techniques. Additionally, comparison with the state-of-the-art demonstrates that the proposed model outperforms the existing models for brain tumor detection.

Keywords: Brain tumor detection; medical imaging; MRI; X-Ray; machine learning; deep learning; majority voting.

Received on January 23, 2024.
Accepted on June 11, 2024.

Introduction

In today's world, the relationship between artificial intelligence and healthcare technologies is increasingly growing, and the advancements in this field are providing significant contributions to our lives through solutions such as semantic web technologies, personalized recommendation systems, intelligent medical solutions, disease diagnosis, and treatment decision support systems (Üzülmez & Cifci, 2024; Çelik Ertuğrul & Elçi, 2020; Cifci, 2022; Babalola et. al, 2021; Çelik Ertuğrul & Celik Ulusoy, 2022; Cifci et.al, 2023; Çelik Ertuğrul et al., 2019; Çelik et al., 2014; Ertuğrul & Abdullah, 2022; Çelik Ertuğrul et al., 2020; Çelik Ertuğrul, 2016; Aktas et al., 2023; Uyguroğlu et al., 2024). These developments are of great importance, especially in the accurate diagnosis and treatment of vital diseases such as brain tumors. Abnormal growth of cells in the brain or nervous system may indicate the presence of a tumor, which can manifest in a variety of shapes and sizes.

Brain tumors are generally classified into two categories: Benign Brain Tumors and Malignant Brain Tumors. Benign tumors, often non-life-threatening, display restricted growth and symptoms, posing minimal harm to surrounding tissues. On the contrary, Malignant tumors pose a higher risk due to their potential for rapid growth, spreading, and damaging neighboring tissues. This can result in a life-threatening condition that significantly affects brain function.

There exist over 120 types of brain tumors, with common examples including Glioblastoma, Meningioma, Metastatic, and Astrocytoma. Early diagnosis and appropriate treatment are crucial. Medical imaging technologies like Magnetic Resonance Imaging (MRI), Computerized Tomography (CT), and X-Ray scans are

integral for early detection in the medical field. However, the time-consuming process of interpreting these scans and communicating results to patients poses potential risks. Figures 1(a) and 1(b) illustrate MRI scans of individuals with and without a brain tumor, respectively.

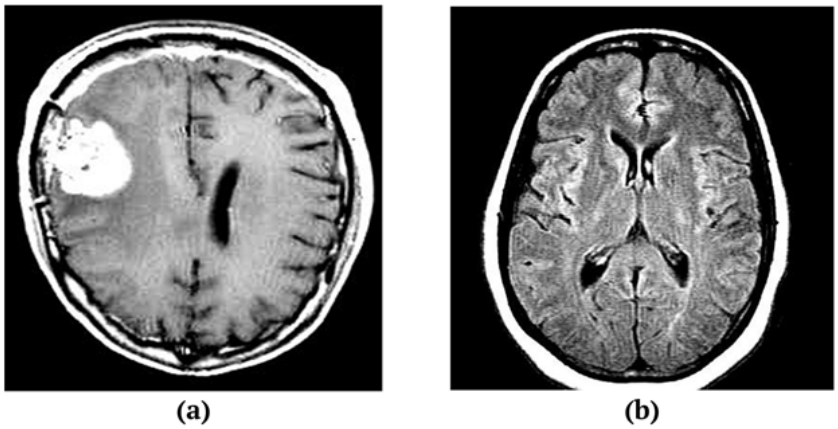

Figure 1. Brain MRI images (a) with brain tumor (b) without brain tumor.

The primary aim of this study is to discern brain tumors through the application of MRI and X-Ray scans, employing a hybrid learning approach with a majority voting mechanism. The potential impact of this research is significant, with the prospect of mitigating life risks through early detection of brain tumors. The utilization of this hybrid learning approach not only facilitates quicker and more cost-effective identification of brain tumors but also offers the advantage of detecting tumors with minimal information, reducing dependencies on additional data. The model developed in this study holds promise for expeditious brain tumor detection, presenting a valuable opportunity to save numerous lives by significantly enhancing the speed of diagnosis.

Symptoms like slurred speech, difficulty speaking, impaired understanding, inappropriate word usage, or speech resembling drunkenness could potentially serve as initial indicators of brain tumors. Detecting these signs, however, can be a time-consuming process, particularly in a bustling hospital setting, where the necessity for comprehensive data collection may impede the timely execution of an effective diagnostic procedure. Nevertheless, for individuals grappling with this undiagnosed condition, every minute holds significant importance in their lives.

The contributions of this paper are as follows. This study endeavors to expedite the detection of brain tumors by integrating deep learning methodologies within a hybrid framework. Employing a dataset consisting of 7600 MRI and X-ray scans, the study trains and tests five prominent deep learning models, namely XGBoost, CNN, ResNet50, DenseNet121, and AlexNet. Throughout the experimental phase, the accuracy, precision, recall, and F1-Score metrics for each model are individually computed and subjected to comparison.

Furthermore, in light of the outcomes derived from these models, a fusion strategy known as the "Majority Voting" approach was implemented to enhance the final performance of the proposed system. The individual performances of the models revealed results of 98.98% for XGBoost, 99.75% for CNN, 99.65% for DenseNet121, 97.21% for ResNet50, and 99.84% for AlexNet. Subsequently, the predictions from all models were fused using the Majority Voting approach. This involves comparing results from all models, and for each test data, the outcome with the highest frequency among the models is selected to form new predictions. The accuracy achieved after employing the Majority Voting approach reached 100.00%.

The rest of the paper is structured as follows: In Section 2, a comprehensive discussion on MRI and X-Ray technologies, the learning approaches utilized in developing and designing the proposed system, and a review of relevant studies in the literature are presented. Section 3 provides intricate details on the open-access datasets used, the applied methodologies, as well as the design, development, and features of the proposed system. Moving on to Section 4, it encompasses the presentation of experimental results, displaying the outcomes of the created deep learning models, followed by the final performance results derived from the hybrid approach. Section 5 is dedicated to the comparison of the results obtained from recent studies utilizing the same MRI/X-Ray datasets in the literature. Furthermore, it includes a comparative analysis with results from recent studies employing different MRI/X-Ray datasets but incorporating similar algorithms as those in

the proposed system. Lastly, the conclusion section provides an overview of the study, offering insights into planned future research endeavors.

Background

This section gives an overview of medical imaging technologies, machine learning and deep learning methods utilized in this study for the classification of brain tumors (Anantharajan et al, 2024; Mikhailova and Anbarjafari, 2022).

Medical imaging technologies

Medical imaging technologies play a crucial role in the diagnosis of brain tumors, with MRI, CT, and X-ray scans being among the most commonly employed methods. Among these, MRI stands out for its ability to provide detailed examinations of organs, tissues, and structures without exposing the patient to radiation. Operating on the principle of Nuclear Magnetic Resonance (NMR), an MRI scanner measures the movement and response of water molecules in the body to a powerful magnetic field. This field aligns hydrogen atoms within the patient's body, and radio waves revitalize these atoms, causing them to reflect the waves.

The scanner captures and converts these reflections into high-quality images, enabling the visualization of various body parts and structures. Figure 2(a) illustrates the setup of an MRI scanner, emphasizing its significance in medical imaging. Structural MRI, employing T1 and T2-weighted imaging, provides detailed insights into the brain's anatomy, while functional MRI (fMRI) serves as another valuable technology for imaging brain function.

The second imaging technology utilized in this study is the X-ray scan, which utilizes X-rays, a form of radiation that can penetrate objects like the human body. Although X-rays are less frequently the primary choice for diagnosing brain cancers, they are beneficial when tumors are located inside or close to bone tissue. In Figure 2(b), an X-ray scanner is depicted, showcasing its application in such scenarios. It is worth noting that due to the lack of X-ray transparency in brain tissue, these scans may not produce detailed images, emphasizing the importance of selecting the appropriate imaging technology based on the specific characteristics of the condition being diagnosed.

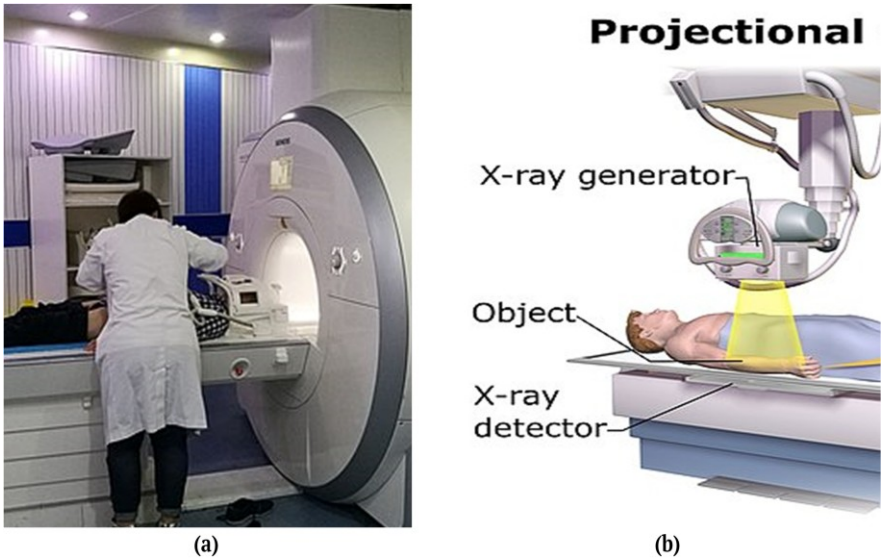

Figure 2. Medical imaging technologies in the diagnosis of brain tumors. (a) MRI scanner and (b) X-Ray scanner.

Deep learning algorithms used

This study employs five learning models: XGBoost, CNN, AlexNet, ResNet50, and DenseNet121. In the subsequent sections, the technical features of each of these models are presented.

Convolutional neural networks (CNN)

The inaugural model employed in this study is the CNN, a formidable type of artificial neural network that has demonstrated remarkable success, particularly in domains like voice recognition and image processing (Li et al., 2021; Rawat & Wang et al., 2020; Liu et al., 2017; Liu et al., 2021). CNNs consist of numerous artificial

neurons or nodes that process input data through distinct layers. As shown in Figure 3, Convolutional, Pooling, and Fully Connected layers constitute the fundamental building blocks of CNN, with the overall architecture depicted in Figure 3(a). Convolution layers operate by applying a sliding operation with filters to images or data matrices, extracting diverse features such as edges, corners, and color nuances. Subsequently, activation layers transform these outputs using nonlinear functions, enhancing the network's ability to discern intricate features. Pooling layers play a vital role in enhancing generalization by reducing dimensionality, thereby alleviating computational burdens. The essence of CNNs lies in their hierarchical conversion of data into multiple features. Initially capturing simple features through filters, the network progressively amalgamates them to form more sophisticated features, imparting deeper semantic understanding. Ultimately, the last layer typically undergoes a classification or regression process to produce the final output. The structural intricacies and functional dynamics of CNNs, as elucidated in the subsequent sections of this study, underscore their pivotal role in the study's analytical framework.

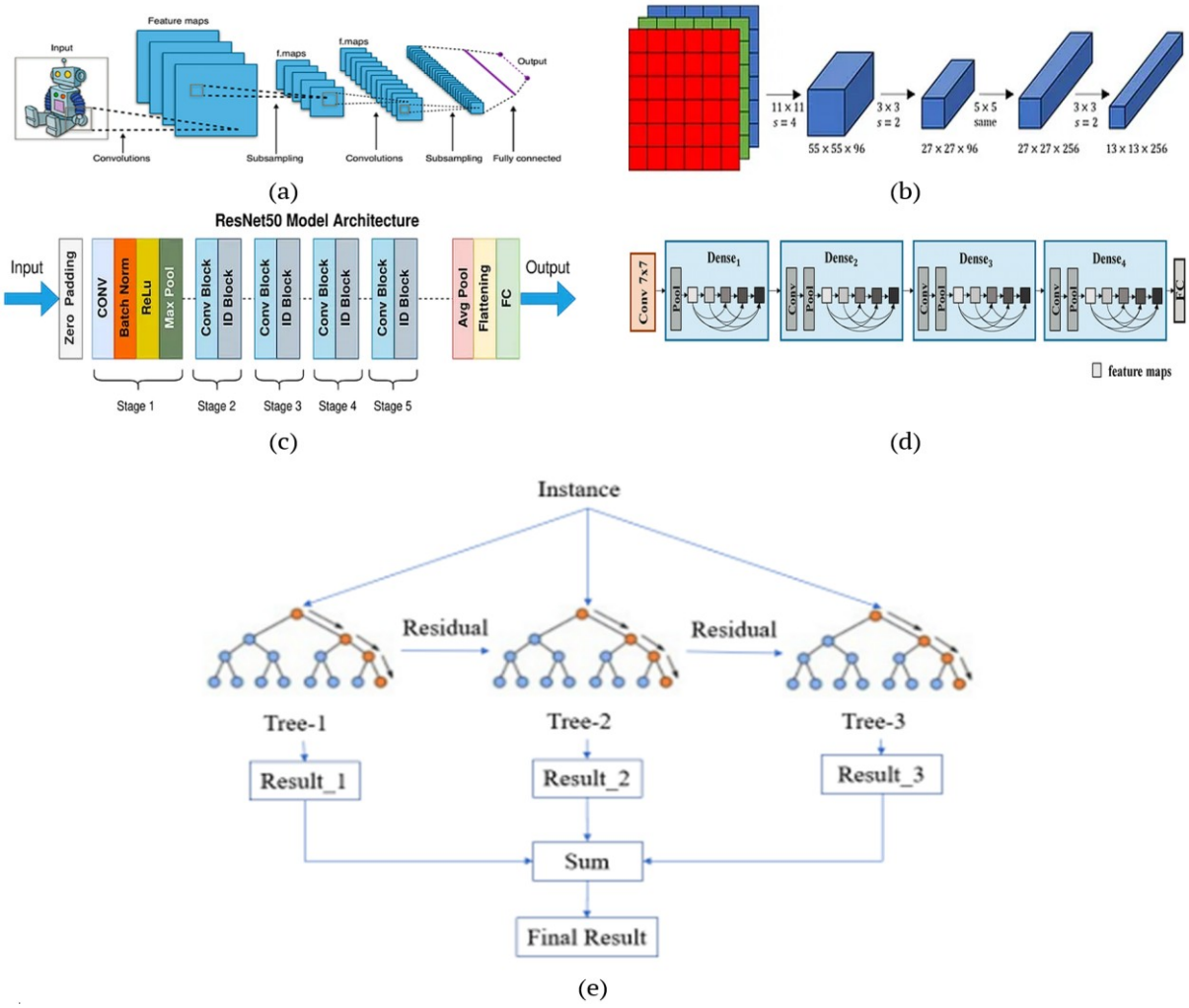

Figure 3. Architectures of the models used. Architectures of (a) CNN (Li et al., 2021), (b) AlexNet (Russakovsky et al., 2015), (c) ResNet50 (He et al., 2016), (d) DenseNet121 (Huang, et al., 2017), and (e) XGBoost (Mitchell et al., 2018).

AlexNet

The second model incorporated into this study is AlexNet, a pioneering deep learning architecture that has ushered in a new era in image classification, representing a significant leap forward for artificial neural networks. Devised in 2012 by Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton, AlexNet garnered widespread acclaim for its groundbreaking success in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) (Russakovsky et al., 2015). Distinguished by its substantially deeper structure compared to earlier classification models, AlexNet has empowered the learning of more intricate features.

The core components of AlexNet predominantly comprise convolutional and pooling layers, strategically designed to sift through data patterns and features hierarchically. This architectural arrangement facilitates

the transition of information from low-level to high-level features. AlexNet also adopts non-linear activation techniques like ReLU (Rectified Linear Activation) to enhance the network's ability to discern complex patterns efficiently. A characteristic feature of AlexNet is its utilization of two GPUs (Graphics Processing Units) for parallel processing, resulting in accelerated computation and training times. This not only enhances efficiency but also contributes to the model's generalizability and mitigates the risk of overfitting through the implementation of dropout techniques. Overfitting occurs when a model exhibits high accuracy on training data but significantly lower accuracy on test data, indicative of memorization rather than learning. For insatance, when a model performs with 99.00% accuracy on the training data but only 66.00% accuracy on the test data after it has been trained, this is considered overfitting since the model remembers the training data rather than learning it. AlexNet's architecture, as illustrated in Figure 3(b), is thoughtfully designed to address and circumvent such challenges, emphasizing its robustness and applicability in diverse image classification tasks within the study.

ResNet50

The third model employed in this study is ResNet50, where "ResNet" stands for Residual Network; He et al., 2016). ResNet is a type of CNN primarily utilized in computer vision applications. ResNet50, specifically, is a formidable 50-layer CNN comprising 48 convolutional layers, 1 MaxPool layer, and 1 average pool layer.

The inception of ResNet, initially with the ResNet34 architecture, aimed to address the vanishing gradient problem encountered when adding more convolutional layers to a CNN. To overcome this challenge, shortcut connections were introduced, enabling the network to skip certain layers. This innovative approach transforms a regular network into a residual network, providing a solution to the vanishing gradient problem. Remarkably, ResNet50 maintains consistent time complexity by doubling the number of filters if the feature map is halved in size.

The architecture of ResNet50 has garnered substantial attention in the field of computer vision due to its exceptional results. Widely adopted in various systems, ResNet50's ability to preserve essential features and mitigate gradient-related challenges has contributed to its prominence. Figure 3(c) visually outlines the architecture of ResNet50, underscoring its intricate design and influential role in advancing the capabilities of CNN within the context of the study.

DenseNet121

The fourth model used in this study is DenseNet121, representing another innovative CNN design. Specifically engineered to enhance the depth, accuracy, and effectiveness of convolutional networks, DenseNet121 features shorter connections between layers, fostering a more interconnected and efficient architecture (Huang et al., 2017). DenseNet121 employs a unique structure where each layer is directly connected to the one preceding it. In contrast to conventional CNNs with L layers having L connections, DenseNet introduces a denser connection pattern, resulting in L(L+1)/2 connections when there are L layers. This dense connectivity, showcased in Figure 3(d), plays a pivotal role in addressing the vanishing gradient problem commonly encountered in traditional CNN architectures (Huang et al., 2017). By ensuring direct connections between layers, DenseNet121 facilitates seamless information flow throughout the network, mitigating gradient-related challenges and contributing to more effective learning. The architecture of DenseNet121, as depicted in Figure 3(d), stands as a testament to its unique design philosophy and its potential impact on advancing the capabilities of CNN (Cui, Chen, & Lu, 2020).

XGBoost

The fifth model used in this study is XGBoost, a formidable machine-learning algorithm renowned for its utilization of gradient boosting techniques (Mitchell et al., 2018). Introduced in 2014, XGBoost has emerged as a robust solution for training and testing large datasets, showcasing exceptional performance in terms of both speed and accuracy. The algorithm's design prioritizes efficiency, allowing it to seamlessly handle vast amounts of data. XGBoost's execution speed is a key feature, tailored to meet the demands of processing extensive datasets efficiently. In terms of performance, XGBoost has demonstrated outstanding results when compared to other gradient boosting algorithms. An additional advantage lies in its open-source nature, making it freely accessible for utilization. The structure of XGBoost, illustrated in Figure 3(e), provides a visual representation of its architecture, highlighting its simplicity and effectiveness in handling diverse machine-learning tasks within the study's context.

Related Works

In this section, we conducted a literature review focused on studies that explore brain tumor detection methodologies utilizing various learning methods and datasets used. As previously mentioned, brain tumors, characterized by abnormal cell growth in the brain or nervous system, pose a significant threat to individuals' life and health. The investigation into diverse learning approaches for detecting and treating brain tumors holds considerable promise. The primary objective of this literature review is to identify existing research studies on brain tumors, discern the different learning techniques used in these studies, and analyze the functionality and impact of these methods. The aim was also to identify notable advancements in brain tumor detection and classification, particularly through the analysis of images acquired from medical imaging devices such as MRI and X-ray, and the integration of various learning algorithms. The subsequent sections will meticulously outline the contributions of the studies reviewed, providing insights into the effectiveness and applicability of these diverse learning methods. Through a systematic examination of the consolidated body of research, this literature review aims to offer a comprehensive understanding of the pivotal role played by various learning approaches in enhancing the accuracy and efficiency of brain tumor detection processes. The findings of this part contribute valuable knowledge to the broader field of medical imaging and diagnostics, encouraging further developments in the understanding and management of brain tumors.

Kuraparthi et al. (2021) proposed a method for detecting brain tumors from MRI images utilizing AlexNet, VGG16 (He et al., 2016), and ResNet50. The study utilized publicly available Kaggle and BRATS datasets, both containing MRI data. The Kaggle dataset comprises a total of 253 MRI images, with 98 belonging to individuals without tumors and 155 belonging to individuals with brain tumors. Meanwhile, the BRATS dataset includes a total of 332 MRI images, with 156 from individuals with lower grade glioma (LGG) and the remaining 176 from individuals with higher grade glioma (HGG). The classification process involved three stages: preprocessing, feature extraction, and classification. In the preprocessing stage, all images were resized differently for each algorithm. For AlexNet, the images were resized to 227x227x3, where the '3' indicates color. For VGG16 and ResNet50, the images were resized to 224x224x3. Data augmentation was employed to enhance the images. Transfer learning and SVM were utilized for classification, resulting in the successful detection of brain tumors from MRI. The study's results indicated that for the Kaggle dataset, VGG16 demonstrated an accuracy of 91.38%, AlexNet obtained 94.83%, and ResNet50 achieved 98.28%. For the BRATS dataset, VGG16 obtained 90.43% accuracy, AlexNet demonstrated 94.68%, and ResNet50 achieved 97.87%.

Tazin et al. (2021) introduced a brain tumor classification method employing CNN. The researchers utilized a publicly available Kaggle dataset, comprising 2513 brain X-ray images of tumors and 2087 X-ray images of healthy brains. Data augmentation techniques were applied to enhance the dataset by increasing its volume. In pursuit of improved results, preprocessing steps were implemented, involving resizing the images to 256x256 and converting them into vectors. For brain tumor detection, the researchers trained MobileNetV2 (Sandler et al., 2018), VGG19 (Simonyan & Zisserman, 2014), and InceptionV3 (Szegedy et al., 2016; Szegedy et al, 2017) models on this dataset. The outcomes revealed that MobileNetV2 achieved an accuracy of 92.00%, VGG19 obtained 88.22%, and InceptionV3 demonstrated 91.00% accuracy. These results demonstrate the effectiveness of the proposed CNN-based method for brain tumor classification, with MobileNetV2 exhibiting particularly high accuracy. The application of various CNN architectures highlights the versatility and adaptability of deep learning techniques in accurately identifying brain tumors from X-ray images.

On the other hand, Khan et al. (2022) presented a method for predicting brain tumors using Transfer Learning. The researchers utilized the Kaggle dataset, which includes 2513 images of individuals with brain tumors and 2087 images of healthy individuals without brain tumors. The dataset incorporates both X-ray and CT scans, with two primary classifications: individuals with brain tumors and healthy individuals. The study leveraged Google Colab for dataset utilization and model training, with the dataset undergoing resizing to 256x256 during the preprocessing stage. Data augmentation techniques were subsequently applied to enrich the dataset. In that research, MobileNetV2 and VGG19 models were trained on the augmented dataset for brain tumor detection. The reported results indicated that MobileNetV2 achieved an impressive accuracy of 97.00%, showcasing its robust predictive capabilities. In contrast, VGG19 attained an accuracy of 91.00%. These findings underscore the efficacy of the proposed Transfer Learning approach, with MobileNetV2 demonstrating notable accuracy in distinguishing between individuals with brain tumors and those without. The utilization of diverse models in the study further highlights the versatility of Transfer Learning techniques in predicting brain tumors from X-ray and CT scans.

In 2022, Ullah et al. (2022a) proposed an innovative approach for the detection and identification of brain tumors utilizing Transfer Learning. The study utilized a Kaggle dataset comprising a total of 3264 MRI images, divided into Training and Testing categories. In the Training set, there were 826 glioma tumor MRI images, 822 meningioma tumor MRI images, 827 pituitary tumor MRI images, and 395 MRI images of individuals without brain tumors. The Testing set included 100 glioma tumor MRI images, 115 meningioma tumor MRI images, 105 pituitary tumor MRI images, and 74 MRI images of individuals without brain tumors. The authors employed nine distinct models, namely InceptionResNetV2 (Szegedy et al, 2016; Szegedy et al, 2017), InceptionV3, Xception (Chollet, 2017), ResNet101, ResNet50, ResNet18, ShuffleNet (Zhang et al., 2018), DenseNet201 (Huang et al., 2017) and MobileNetV2, respectively. These models were trained on their dataset. During testing with the dataset, the InceptionResNetV2 model exhibited the highest accuracy, achieving an impressive score of 98.91%. On the other hand, ResNet18 model achieved the lowest accuracy score of 67.03%, while ResNet101 model demonstrated an accuracy score of 74.09%. These outcomes emphasize the robust performance of Transfer Learning techniques, particularly with the superior accuracy achieved by the InceptionResNetV2 model, showcasing its potential for accurate brain tumor detection and identification from MRI images.

Moirangthem, Singh, and Singh (2021) introduced a novel method for image classification and retrieval framework focused on brain tumor detection using CNN on Region of Interest (ROI). The researchers utilized a dataset obtained from Kaggle, comprising brain tumor MRI images divided into two categories: 'yes' and 'no.' The 'yes' category included MRI scans of individuals with brain tumors (155 images), while the 'no' category consisted of MRI scans of healthy individuals without brain tumors (98 images). In the preprocessing stage, the dataset was resized to 224x224, and grayscale images were converted to RGB. The study employed three models for brain tumor detection: Simple CNN, ResNet50, and ResNet50 with ROI. According to the reported results, the Simple CNN model achieved an accuracy score of 62.00%. The ResNet50 model exhibited an improvement with an accuracy score of 82.00%. Notably, the ResNet50 with ROI model surpassed both, achieving the highest accuracy score of 87.00%. These findings highlight the effectiveness of incorporating ROI in conjunction with advanced CNN architectures like ResNet50, showcasing the potential for accurate brain tumor detection in MRI images.

On the other hand, in 2020, Amin et al. (2020) proposed a distinctive approach for brain tumor detection from MRI, leveraging multiple publicly available datasets. One dataset sourced from Nashtar Hospital Multan included 46 MRI datasets from individuals with brain tumors and 39 MRI datasets from healthy individuals. The Harvard dataset comprised 65 MRI datasets from people with brain tumors and 35 MRI datasets from people without brain tumors. Additionally, the RIDER dataset featured MRI datasets from individuals with brain tumors at different stages, categorized as grades 1 through 4, with 36 grade-1, 41 grade-2, 26 grade-3, and 23 grade-4 brain tumor MRI datasets. The dataset underwent a preprocessing stage, involving the extraction of skull, background, eyes, and scalp from the images to eliminate unnecessary features that might affect predictions.

The images were then converted to grayscale, and a Gaussian filter was applied. For classification, the study employed an SVM classifier trained on these datasets using cross-validation. The reported results for each dataset are as follows: Nashtar Hospital Multan dataset: 98.80% accuracy (ACC), 0.98 area under the curve (AUC), 96.90% sensitivity, and 100.00% specificity. Harvard dataset: 99.40% accuracy, 1.00 AUC, 98.40% sensitivity, and 100.00% specificity. RIDER dataset: 100.00% accuracy, 1.00 AUC, 100.00% sensitivity, and 100.00% specificity. These scores emphasize the robust performance of the proposed approach, showcasing high accuracy and sensitivity across different datasets, indicating its potential for effective brain tumor detection from MRI images.

Ullah et al. (2022b) proposed an End-to-End Deep Learning-based approach for effective and reliable brain tumor detection. The study utilized a Kaggle dataset comprising a total of 3060 MRI images, categorized into “yes” and “no” groups. The “yes” category included MRI images of 1500 individuals with brain tumors, while the “no” category contained MRI images of 1500 healthy individuals. Additionally, a “pred” folder in the dataset was designated for test data, featuring MRI images of 60 individuals with brain tumors and healthy individuals. The authors employed the TumorResNet model, a variant of ResNet18 architecture with the addition of the Leaky Rectified Linear Unit (LReLU) activation function. The dataset was split into 80% for training and 20% for testing. With the proposed method, the study achieved impressive results, including 99.33% accuracy, 99.50% precision, 99.50% recall, 100.00% specificity, and a 99.50% F1-score. These

outcomes underscore the effectiveness and reliability of the End-to-End Deep Learning approach using the TumorResNet model for accurate brain tumor detection from MRI.

Molachan et al. (2021) proposed a method for brain tumor detection using CNN on MRI images (Olachan, Manoj, & Dhas, 2021). The study utilized a Kaggle dataset comprising a total of 253 MRI images, categorized into "yes" and "no" categories. The "yes" category included MRI images of 155 individuals with brain tumors, while the "no" category contained MRI images of 98 healthy individuals. Due to the limited size of the original dataset, the authors applied data augmentation to increase the volume. After data augmentation, a total of 2065 MRI images were obtained, comprising 1085 images with brain tumors and 980 without. The augmented data was split into 70% for training and 30% for testing during the model experimentation phase. The authors employed a CNN model for their study. Subsequently, they compared the performance of their model with the VGG16 model. The proposed model achieved an accuracy score of 91.20%, demonstrating its effectiveness in detecting brain tumors from MRI images, even when compared to a well-established model like VGG16.

Table 1 compares various brain tumor detection methodologies and similar studies in the literature using different learning techniques and datasets.

Table 1. Comparison table for similar studies from literature.

Study	Datasets	Methods	Preprocessing Methodologies	Results Obtained	Notes
Kuraparthi et al. (2021)	Kaggle (253 MRI, 98 healthy, 155 brain tumor), BRATS (332 MRI, 156 lower grade glioma, 176 higher grade glioma)	AlexNet, VGG16, ResNet50	Image resizing, data augmentation	Kaggle: VGG16: 91.38%, AlexNet: 94.83%, ResNet50: 98.28% BRATS: VGG16: 90.43%, AlexNet: 94.68%, ResNet50: 97.87%	Three-stage classification (preprocessing, feature extraction, classification), SVM used.
Tazin et al. (2021)	Kaggle (2513 brain X-ray tumors, 2087 healthy brain X-rays)	MobileNetV2, VGG19, InceptionV3	Image resizing (256x256), data augmentation	MobileNetV2: 92.00%, VGG19: 88.22%, InceptionV3: 91.00%	CNN-based method, high accuracy in X-ray images.
Khan et al. (2022)	Kaggle (2513 X-ray and CT tumors, 2087 healthy X-ray and CT)	MobileNetV2, VGG19	Image resizing (256x256), data augmentation	MobileNetV2: 97.00%, VGG19: 91.00%	Transfer Learning, high accuracy in X-ray and CT images.
Ullah et al. (2022a)	Kaggle (3264 MRI, 826 glioma, 822 meningioma, 827 pituitary, 395 healthy)	InceptionResNetV2, InceptionV3, Xception, ResNet101, ResNet50, ResNet18, ShuffleNet, DenseNet201, MobileNetV2	Data augmentation, data splitting (training/testing)	Highest accuracy: InceptionResNetV2 98.91%, Lowest accuracy: ResNet18 67.03%	High accuracy with Transfer Learning in MRI images.
Moirangthem et al. (2021)	Kaggle (155 MRI tumors, 98 healthy MRIs)	Simple CNN, ResNet50, ResNet50 with ROI	Image resizing (224x224), grayscale to RGB conversion	Simple CNN: 62.00%, ResNet50: 82.00%, ResNet50 with ROI: 87.00%	CNN-based method with ROI incorporation, significant improvement in accuracy.
Amin et al. (2020)	Nashtar Hospital Multan (46 brain tumor, 39 healthy), Harvard (65 brain tumor, 35 healthy), RIDER (36 grade 1 tumor, 42 grade 2 tumor, 26 grade 3 tumor, 23 grade 4 tumor)	SVM	Skull, background, eyes, and scalp extraction, grayscale conversion, Gaussian filter	Nashtar: Accuracy: 98.80%, AUC: 0.98, Sensitivity: 96.90%, Specificity: 100.00% Harvard: Accuracy: 99.40%, AUC: 1.00, Sensitivity: 98.40%, Specificity: 100.00% RIDER: Accuracy: 100.00%, AUC: 1.00, Sensitivity: 100.00%, Specificity: 100.00%	High accuracy and sensitivity across multiple datasets, effective brain tumor detection from MRI images.

Ullah et al. (2022b)	Kaggle (3060 MRI, 1500 brain tumor, 1500 healthy)	TumorResNet (ResNet18 + LReLU)	Image resizing (224x224)	Accuracy: 99.33%, Precision: 99.50%, Recall: 99.50%, Specificity: 100.00%, F1 Score: 99.50%	End-to-End Deep Learning method with high accuracy in MRI images.
Molachan et al. (2021)	Kaggle (253 MRI tumors, 98 healthy MRIs)	CNN, VGG16	Data augmentation	CNN: 91.20% accuracy	Comparison with VGG16, effective brain tumor detection from MRI images.

Materials and methods

Datasets Used

In this study, two datasets are used, both freely and publicly available on the Kaggle website. The first dataset, named Dataset 1, comprises X-ray images with varying dimensions (Viradiya & X-Ray Brain Tumor Dataset, 2021). Dataset 1 is classified into two categories: "Brain Tumor" and "Healthy" patients. It contains a total of 2087 X-ray images of healthy individuals and 2513 X-ray images of individuals with brain tumors. The files in Dataset 1 have extensions of both ".tif" and ".jpg", with a dataset size of 112 MB.

The second dataset, named Dataset 2, consists of MRI images with diverse dimensions (Panigrahi, 2021). This dataset also includes a separate folder containing mixed MRI images of individuals with brain tumors and healthy individuals for prediction purposes. Dataset 2 comprises MRI images from 1500 individuals with brain tumors and 1500 healthy individuals, with the total size of the dataset being 67 MB. The images in this dataset are presented in jpg format. Table 2 demonstrates the number of healthy and tumor images for all datasets used in this study.

Table 2. Number of images in the datasets used.

Dataset	Reference	Healthy	Tumor	Total
Dataset 1	(Viradiya, 2021)	2087	2513	4600
Dataset 2	(Panigrahi, 2021)	1500	1500	3000
Dataset 1 + Dataset 2	-	3587	4013	7600

System architecture and user interactions

System design and model

Figure 4 illustrates the architecture diagram of the proposed hybrid learning system with the majority voting. The model developed in this study encompasses three key stages: Preprocessing, Matching, and Classification. As depicted in Figure 4, the system initially requires MRI and X-Ray images as test data to be employed by all previously established models for generating predictions.

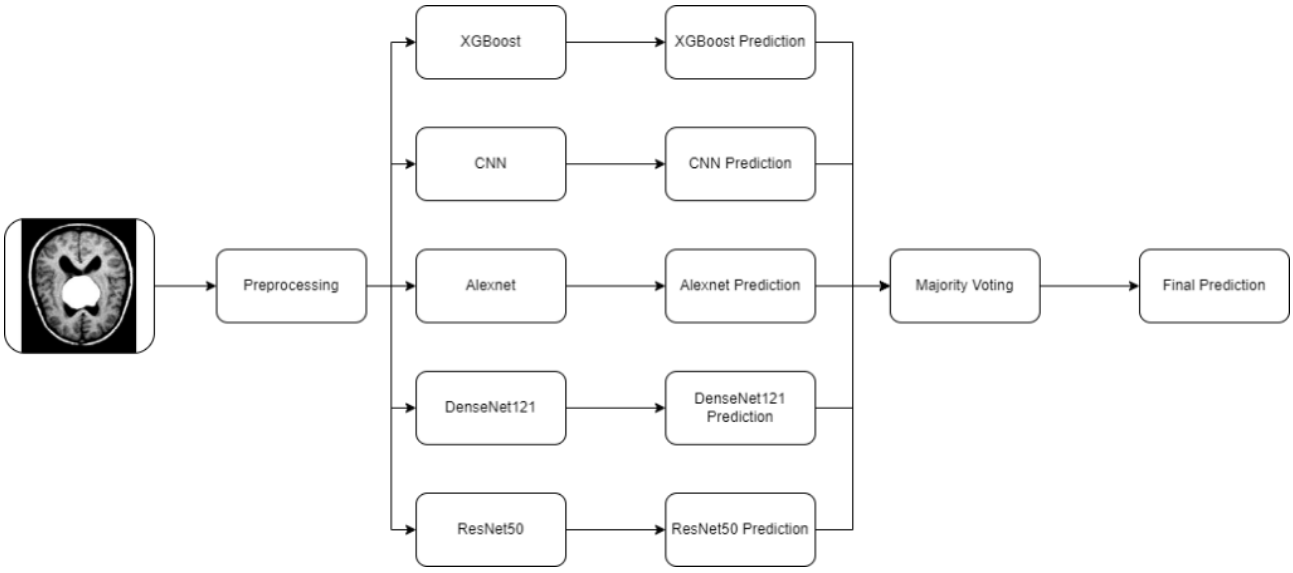

Figure 4. Block diagram of proposed method.

The models generate outputs based on the provided test data. Subsequently, an aggregate output is formed by applying the Majority Voting method to these individual estimates, resulting in the final prediction. This approach integrates multiple models to enhance the robustness and reliability of the overall system.

The initial step involves a preprocessing stage to utilize the collected data effectively. During this stage, images are resized to a uniform size of 128x128x3. Additionally, labels are assigned to indicate the respective categories each image belongs to. Subsequently, deep learning models are trained using these standardized images. Five distinct models—XGBoost, CNN, ResNet50, DenseNet121, and AlexNet—are trained in this phase. Once all models have been trained and tested, the Majority Voting approach is implemented. Through Majority Voting, the outputs from each model are fused in the matching stage, and this collective result is considered as the output of the overall model. Majority Voting enhances reliability and accuracy, providing a robust model.

For a more in-depth understanding of the methodologies applied in this study, a comprehensive overview of the experimental process is presented in Figure 5. In Phase 1, datasets including Dataset 1, Dataset 2, and the combined Dataset 1 + Dataset 2 are acquired. Due to variations in image dimensions in the Dataset 1 and Dataset 2, Phase 2 involves resizing these images to a standardized 128x128 format. Phase 3 results in obtaining resized and RGB images. Subsequently, in Phase 4, each dataset is split into 80% for training and 20% for testing. Phase 5 utilizes the 80% training data to train models, including XGBoost, AlexNet, CNN, DenseNet121, and ResNet. Phase 6 tests the models using Dataset 1 with the 20% test data allocated from Dataset 1. Similarly, Phase 7 tests models created using Dataset 2 with 20% test data separated from Dataset 2. In Phase 8, models created with the combined Dataset 1 + Dataset 2 are tested with 20% test data from this combined dataset. Moving to Phase 9, the Majority Voting method is applied to the models obtained from Dataset 1 + Dataset 2. Phase 10 involves taking predictions made by these models on the combined Dataset 1 + Dataset 2, creating a new prediction matrix by selecting the most frequent predictions for each test data. In Phase 11, ground truth is used for each patient's correct result in the test data. Phase 12 involves comparing the predictions obtained from the Majority Voting method with the ground truths, and performance metrics including Accuracy, Specificity, Sensitivity, Precision, Recall, and F1-score are calculated. Phase 13 marks the completion of this experimental process.

Implementation

Development tools used

In this research, we employed the Python programming language for developing models, data acquisition, and preprocessing. Python stands out as a leading language in contemporary software development, data analysis, and applications in Machine Learning and Artificial Intelligence. Additionally, Python is flexible, allowing the creation of websites and desktop applications. Its status as an easy-to-learn language is due to its straightforward syntax, offering a quick development opportunity. Our preference for Python in this study was twofold. First, it boasts a rich set of libraries conducive to model development. Second, it provides the flexibility to develop in a cloud environment, specifically on Google Colab. Google Colab Pro served as the platform for constructing and compiling models. It is a free, high GPU capacity cloud service tailored for developing deep learning applications. This service facilitates the use of deep learning libraries like TensorFlow, Keras, OpenCV, as well as programming languages such as Python and R. The substantial GPU capacity ensures efficient and rapid model training. Additionally, Google Colab seamlessly integrates with Google Drive, enabling all developments to occur in the cloud.

Methodologies applied

Pre-processing

Before training and testing the data, an important step of "pre-processing" is applied. In this phase, the data for all models is uniformly resized to 128x128x3 dimensions, where '3' denotes the RGB color channels in the image. The resizing is essential because images often vary in size, and models generally perform better with standardized, smaller images. Subsequently, these images are transformed into vector form. It is noteworthy that a specific adjustment is made for the XGBoost model. Unlike other models, the XGBoost model processes training images in an inverted matrix form. Since the XGBoost model takes training images as an inverted matrix compared to other models, the images for XGBoost are reshaped. Reshaping was not applied for other models.

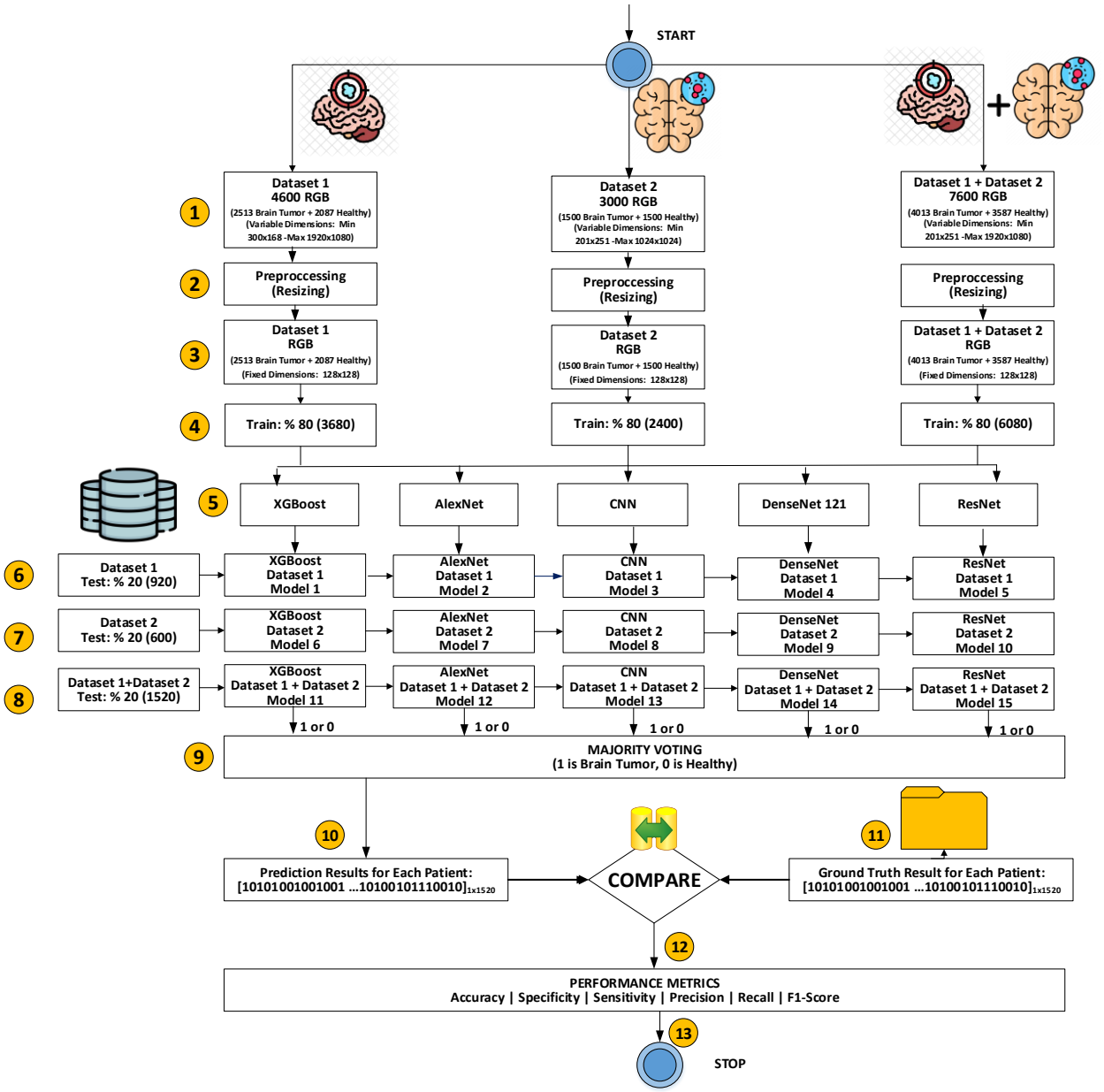

Figure 5. Detailed Progress of Experimental Studies.

Stage-1: creating XGBoost model

The first model that is created is XGBoost model. To start implementing the model, the data on Google Drive was first accessed on Google Colab. Preprocessing was applied on this accessed data. Then, these data are split into 80% training and 20% test data. When training this model, unlike other models, the given data should be reshaped as a 2D matrix. Also, hyper parameters are very important when training this model. These hyperparameters are "n_estimators", "max_depth", and "learning_rate" respectively. Among these parameters, "n_estimators" is set to 100. This parameter represents how many trees will be generated and generally improves performance, but if it is increased too much, it will cause overfitting. The second parameter is "max_depth". This parameter sets the depth of the generated trees. Increasing this parameter usually improves performance but can cause overfitting and can also cause crashes as it increases memory usage too much. This parameter is set to 5. The other parameter is "learning_rate". This parameter represents the learning rate. It should be given a value between 0 and 1 and the closer to 0 the faster it learns. This parameter is set to 0.1 in the model. K-fold cross validation was applied as 3-fold on this model. After training this model, it was tested with test data and performance metrics of the model such as confusion matrix, accuracy, sensitivity, specificity, f1-score are given as output. Then the model is saved to Google Drive. All of these processes are applied for the first dataset, the second dataset and the dataset consisting of a mixture of these two datasets.

Stage-2: creating CNN model

The second model created is CNN. When started to implement this model, our dataset is accessed on Google Drive via Google Colab. Preprocessing was applied to this accessed data. Then this data was split into 80% training and 20% test data. The model was trained with the data that is allocated for training. While training this model, some hyperparameters are changed on the model. The most important of these is the "learning_rate" parameter and this parameter is set to 0.00001. This parameter controls the learning rate of the model. This model was run for 10 epochs. Afterwards, the performance of the model was tested with the data allocated for testing and performance metrics such as confusion matrix, accuracy, sensitivity, specificity, F1-score are obtained as output. Then, the model created is saved to Google Drive. All of these processes are applied for the first dataset, the second dataset and the dataset consisting of a mixture of these two datasets.

Stage 3: creating ResNet50 model

The third model created is ResNet50. When started to implement this model, our dataset is accessed on Google Drive via Google Colab. Preprocessing was applied to this accessed data. Then this data was split into 80% for training and 20% for testing. While training this model, some hyperparameters were changed on the model. The most important of these is the "learning_rate" parameter and this parameter is set to 0.00001. This parameter controls the learning rate of the model. This model is run for 20 epochs. The model is trained with the data that is allocated for training. Afterwards, the performance of the model is tested with the data that is allocated for testing and performance metrics such as confusion matrix, accuracy, sensitivity, specificity, f1-score are obtained as output. Then, model created is saved. All of the processes are applied for the first dataset, the second dataset, and the dataset consisting of a mixture of these two datasets.

Stage 4: creating AlexNet model

The fourth model that is created is AlexNet. When started to implement this model, our dataset is accessed on Google Drive via Google Colab. Preprocessing is applied to this accessed data. Then the data is split into 80% for training and 20% for testing. The model is trained with the data that is allocated for training. While training this model, 5-fold K-fold cross validation is applied. For each fold, the model is run for 20 epochs. Afterwards, the performance of the model is tested with the data that is allocated for testing and performance metrics such as confusion matrix, accuracy, sensitivity, specificity, f1-score is obtained as output. All of these processes are applied for the first dataset, the second dataset and the dataset consisting of a mixture of these two datasets.

Stage 5: creating DenseNet121 model

The fifth model that is created is DenseNet121. When started to implement this model, our dataset is accessed on Google Drive via Google Colab. Preprocessing is applied to this accessed data. Then this data was split into 80% for training and 20% for testing data. The model is trained with the data that is allocated for training. When training this model, the standard hyperparameters are not changed much, only the number of epochs is changed, and the model is trained for 15 epochs. Afterwards, the performance of the model is tested with the data that is allocated for testing and performance metrics such as confusion matrix, accuracy, sensitivity, specificity, f1-score are given as output. Then, the model that is created is saved to Google Drive. All of the processes are applied for the first dataset, the second dataset and the dataset consisting of a mixture of these two datasets.

Stage 6: applying majority voting (MV) method

As the last step, the Majority Voting method is applied to the models that are created. For this stage, all created models are called from Google Drive. Then, to get the predictions of all models, the dataset from Google Drive is called and fixed test data is created. Predictions are made with the models that is called on this test data. Predictions made for each model is obtained. Then predictions of each model are taken, the prediction that is selected as the majority in our five models is found and this prediction is added to the prediction matrix created. In this way, a system that is reliable and provides high accuracy is obtained.

Experimental studies

Performance metrics such as accuracy, sensitivity, specificity, precision, recall, F1-score are used to measure the performance of the models used. The measures to be used when calculating these metrics are True Positive (TP), True Negative (TN), False Positive (FP), False Negative (FN). TP refers to the number of brain tumor data detected as brain tumors. TN refers to the number of healthy data that were detected as healthy. FP refers to the number of brain tumor data detected as healthy. FN refers to the number of healthy data detected as brain tumors. The following equations (1) to (6) are used to calculate the following performance metrics.

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

$$Sensivity = \frac{TP}{TP+FN} \quad (2)$$

$$Specifity = \frac{TN}{TN+FP} \quad (3)$$

$$Precision = \frac{TP}{TP+FP} \quad (4)$$

$$Recall = \frac{TP}{TP+FN} \quad (5)$$

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision+Recall} \quad (6)$$

As shown in Figure 5, the experimental progress planned is applied on 7600 MRI and X-Ray scans that are trained and tested using the deep learning models such as XGBoost, CNN, ResNet50, DenseNet121 and AlexNet. The TP, TN, FP, and FN results for each model on different datasets are shown in the Figure 6. The values of TP, TN, FP, and FN generated for each model, utilizing Dataset 1, Dataset 2, and Dataset 1 + Dataset 2 individually, as ascertained from our experimental investigations, are enumerated below.

For dataset-1 + dataset-2 without MV:

— ***XGBoost:*** Achieved a true positive rate of 99.59%, a false positive rate of 0.41%, a false negative rate of 0.44%, and a true negative rate of 99.56%.

— ***CNN***: Demonstrated a true positive rate of 99.70%, a false positive rate of 0.30%, a false negative rate of 0.21%, and a true negative rate of 99.79%.

— ***AlexNET***: Attained a true positive rate of 99.80%, a false positive rate of 0.20%, a false negative rate of 0.22%, and a true negative rate of 99.78%.

— ***DenseNet121***: Recorded a perfect true positive rate of 100.00%, with a false positive rate of 0.00%, a false negative rate of 0.72%, and a true negative rate of 99.28%.

— ***ResNet50***: Demonstrated a perfect true positive rate of 100.00%, with a false positive rate of 0.00%, a false negative rate of 5.68%, and a true negative rate of 94.32%.

For dataset-1 + dataset-2 With MV:

— ***Majority Voting:*** Achieved a perfect true positive rate of 100%, with no false positives, no false negatives, and a true negative rate of 100.00%.

Subsequently, the TP, TN, FP, and FN results obtained from each model, utilizing Dataset 1, Dataset 2, and Dataset 1 + Dataset 2 individually, serve as the basis for delineating the performance outcomes as expounded in the next section. These results are also visually represented in Figure 6.

The results obtained from Dataset 1

For Dataset 1, accuracy, sensitivity, specificity, precision, recall, F1-score results obtained are shown on Table 3. All models demonstrate strong overall performance, with accuracy ranging from 96.95% to 99.35%. Sensitivity, specificity, precision, and recall scores are generally high across models. DenseNet121 stands out as the top-performing model, achieving the highest accuracy and demonstrating excellent sensitivity, specificity, precision, and recall. CNN performs well with a particularly high sensitivity but at the cost of somewhat lower specificity. XGBoost shows balanced performance with good accuracy, sensitivity, specificity, precision, and recall. ResNet50 exhibits strong specificity and precision but has a lower sensitivity compared to other models. AlexNet performs reasonably well but has a slightly lower sensitivity.

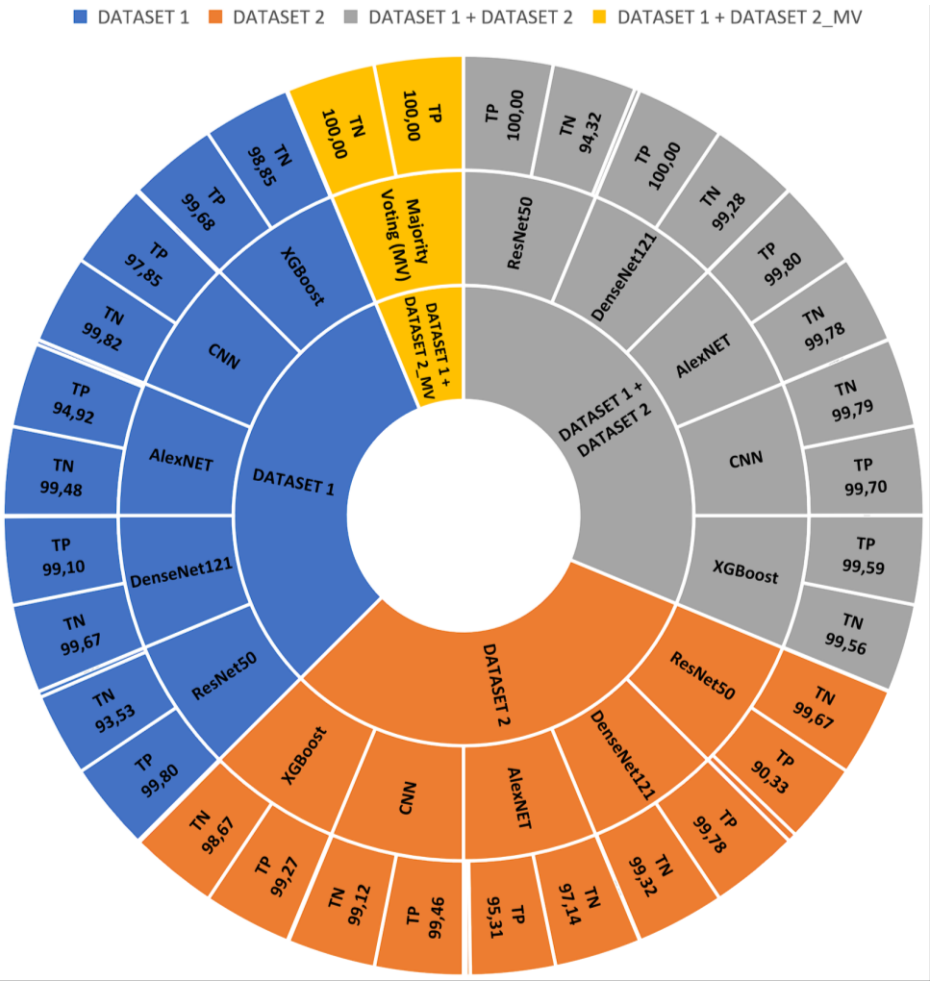

Figure 6. Entire TP, TN, FP, and FN results obtained from each model applied on Dataset 1, Dataset 2, Dataset 1+ Dataset 2 Without MV, and Dataset 1+ Dataset 2 With MV.

Table 3. The results obtained with the models on Dataset 1.

Model	Accuracy(%)	Sensitivity	Specificity	Precision	Recall	F1-score
XGBoost	99.30	0.988	0.996	0.996	0.988	0.99
CNN	98.83	0.998	0.978	0.983	0.998	0.99
AlexNet	97.60	0.96	0.99	0.99	0.96	0.97
DenseNet121	99.35	0.996	0.99	0.99	0.99	0.99
ResNet50	96.95	0.95	0.997	0.998	0.95	0.97

The results obtained from Dataset 2

The models developed using Dataset 2 yielded the performance results shown in Table 4. DenseNet121 again stands out as the top-performing model, achieving the highest accuracy and demonstrating excellent sensitivity, specificity, precision, recall, and F1-score. CNN performs well with high accuracy and balanced sensitivity and specificity. XGBoost shows balanced performance with good accuracy, sensitivity, specificity, precision, recall, and F1-score. ResNet50 exhibits a trade-off between sensitivity and specificity, resulting in a slightly lower accuracy compared to other models. AlexNet performs reasonably well but has a lower accuracy and slightly unbalanced sensitivity and specificity.

Table 4. The obtained results with the models on Dataset 2.

Model	Accuracy	Sensitivity	Specificity	Precision	Recall	F1-score
XGBoost	98.97	0.987	0.99	0.99	0.987	0.989
CNN	99.27	0.99	0.994	0.995	0.99	0.993
AlexNet	96.2	0.97	0.95	0.95	0.944	0,946
DenseNet121	99.55	0.993	0.997	0.997	0.993	0.994
ResNet50	95.00	0.996	0.903	0.903	0.996	0.95

The results obtained from the mixture of two datasets

The models created using the results obtained on the dataset formed by mixing the two datasets are shown in Table 5. Majority Voting stands out as the top-performing model, achieving perfect scores across all metrics. DenseNet121 and AlexNet also demonstrate exceptional performance with high accuracy and balanced sensitivity and specificity. CNN follows closely, excelling in sensitivity and achieving a very high F1-score. XGBoost and ResNet50 show solid performance, but with a slightly lower accuracy compared to other models.

Table 5. The results obtained with the models on Dataset 1 + Dataset 2.

Model	Accuracy	Sensitivity	Specificity	Precision	Recall	F1-score
XGBoost	98.98	0.984	0.995	0.995	0.985	0.99
CNN	99.75	0.998	0.997	0.997	0.998	0.997
AlexNet	99.84	0.998	1	0.998	0.997	0,998
DenseNet121	99.65	0.993	1	1	0.993	0 .996
ResNet50	97.21	0.94	1	1	0.94	0,97
Proposed Method	100.00	1	1	1	1	1

Table 6 shows the execution times for training and testing of each algorithm that is used. XGBoost exhibits a considerably shorter training execution time compared to the deep learning models. Among the deep learning models, AlexNet has the shortest training execution time. For testing, all models, including the deep learning ones, have very short execution times. XGBoost has an exceptionally low-test execution time. DenseNet121 has the longest training execution time among the deep learning models. ResNet50 has a longer test execution time compared to other models. Individual models of the system are trained sequentially in the experiments and the decisions obtained from each model are used for the fusion of the decisions in the proposed system. Naturally, it is expected for the proposed method to be executed considering the summation of all individual models' execution times if the models are executed sequentially. Therefore, the execution time of the proposed method is longer than that of the individual systems. However, the performance of the proposed method is increased to 100% when we combine five different methods by taking the strengths of different individual sytems. In that case, fusion of multiple methods boost the proposed method to classify the brain tumors perfectly which is the most important issue for health systems.

Table 6. Execution times of algorithms.

Model	Train Execution Time(s)	Test Execution Time(s)
XGBoost	1275.06	0.004522
CNN	2004.92	0.088054
AlexNet	1209.92	0.078618
DenseNet121	3193.91	0.123197
ResNet50	1190.92	0.071975
Proposed Method	8874.73	0.373905

The choice of a model may depend not only on accuracy but also on the computational resources available and the acceptable execution times. XGBoost is computationally efficient for training but may not capture complex patterns as effectively as deep learning models. Deep learning models (CNN, AlexNet, DenseNet121, ResNet50) generally have longer training times but provide powerful representation for learning complex tasks.

Comparison with the state-of-the-art

This section presents comparison with the state-of-the-art. Table 7 shows our study results and the results obtained in other studies using the same dataset as ours. As seen in Table 7, there are significant differences between the results obtained in other studies on the same dataset and the results obtained from our study, the results obtained in our study are better than the results obtained from the state-of-the-art except ResNet50. Since no similar study using Dataset 2 was found in the literature, only the comparison results of Dataset 1 can be compared and given in this section.

Table 8 shows the comparison of our results with the results from similar studies using the same method but with different datasets. The comparison of similar studies in the literature working on brain tumor prediction using MRI images with our study, according to the methods used, names of the datasets, dataset

sizes, image resizing methodology, dimensions of the training/test datasets, and the accuracy, sensitivity, recall and F1-score achieved are summarized in Table 8.

Table 6. Comparison of our results with other studies using different models on the Dataset 1.

Authors	Year	Configuration	İmage Resize	Dataset #	F1-Score (%)	Accuracy (%)
Tazin et al.	2021	VGG19	256 × 256	Dataset 1	88.18	88.22
Tazin et al.	2021	InceptionV3	256 × 256	Dataset 1	90.98	91.00
Tazin et al.	2021	MobileNetV2	256 × 256	Dataset 1	92.00	92.00
Khan et al.	2022	VGG19	256 × 256	Dataset 1	91.00	91.00
Khan et al.	2022	MobileNetV2	256 × 256	Dataset 1	97.00	97.33
Our Study	2024	XGBoost	128 x 128	Dataset 1	99.20	99.30
Our Study	2024	CNN	128 x 128	Dataset 1	99.00	98.83
Our Study	2024	Alexnet	128 x 128	Dataset 1	97.00	97.60
Our Study	2024	DenseNet121	128 x 128	Dataset 1	99.00	99.35
Our Study	2024	ResNet50	128 x 128	Dataset 1	97.00	96.95
Our Study	2024	Proposed Method	128 x 128	Dataset 1	100.00	100.00

Table 8. Comparison of our results with the state-of-the-art using the same methods on different datasets.

Authors	Year	Configuration	Dataset Name	Data Size	Image Resize	Train/Test (%)	Accuracy (%)	Precision	Recall	F1-Score (%)
Kuraparthi et al.	2021	VGG16	Kaggle	253	224x224	70/30	91.38	86.66	92.86	91.23
Kuraparthi et al.	2021	AlexNet	Kaggle	253	227x227	70/30	94.83	93.10	92.86	94.74
Kuraparthi et al.	2021	ResNet50	Kaggle	253	224x224	70/30	98.28	96.55	100.00	98.25
Kuraparthi et al.	2021	VGG16	BRATS	332	224x224	70/30	90.43	89.36	91.30	90.32
Kuraparthi et al.	2021	AlexNet	BRATS	332	227x227	70/30	94.68	93.62	95.65	94.62
Kuraparthi et al.	2021	ResNet50	BRATS	332	224x224	70/30	97.87	95.74	100.00	97.83
Ullah et al.	2022a	ResNet101	Kaggle - Brain Tumor Classification (MRI)	3264	224x224	80/20	74.09	73.19	67.23	70.08
Ullah et al.	2022a	ResNet18	Kaggle - Brain Tumor Classification (MRI)	3264	224x224	80/20	63.04	64.63	52.09	57.68
Ullah et al.	2022a	InceptionResNetV2	Kaggle - Brain Tumor Classification (MRI)	3264	299x299	80/20	98.91	98.28	99.75	99.00
Ullah et al.	2022a	Inceptionv3	Kaggle - Brain Tumor Classification (MRI)	3264	229x229	80/20	94.48	93.00	94.50	93.74
Ullah et al.	2022a	Xception	Kaggle - Brain Tumor Classification (MRI)	3264	224x224	80/20	98.37	98.51	99.25	98.87
Ullah et al.	2022a	ShuffleNet	Kaggle - Brain Tumor Classification (MRI)	3264	224x224	80/20	89.31	87.96	87.43	87.69
Ullah et al.	2022a	DenseNet201	Kaggle - Brain Tumor Classification (MRI)	3264	224x224	80/20	68.71	73.04	67.46	70.14
Ullah et al.	2022a	ResNet50	Kaggle - Brain Tumor Classification (MRI)	3264	224x224	80/20	67.03	70.55	68.13	69.32
Ullah et al.	2022a	MobileNetV2	Kaggle - Brain Tumor Classification (MRI)	3264	224x224	80/20	67.03	70.55	68.13	69.32
Amin et al.	2017	SVM	Harvard	100	-	80/20	99.4	-	-	-

Amin et al.	2017	SVM	RIDER	126	-	80/20	100.00	-	-	-
Amin et al.	2017	SVM	Nashtar Hospital Multan	85	-	80/20	98.80	-	-	-
Ullah et al.	2022b	Tumor Resnet	Kaggle BTD-MRI	3000	224x224	80/20	99.33	99.50	99.50	99.50
Molachan et al.	2021	CNN	Kaggle - Brain MRI Images for Brain Tumor Detection	253	-	70/30	91.20	-	-	-
Our Study	2023	XGBoost	Dataset 1 + Dataset 2	7600	128x128	80/20	98.98	99.50	98.50	99.00
Our Study	2023	CNN	Dataset 1 + Dataset 2	7600	128x128	80/20	99.75	99.70	99.80	99.80
Our Study	2023	AlexNet	Dataset 1 + Dataset 2	7600	128x128	80/20	99.84	99.80	99.70	99.80
Our Study	2023	DenseNet121	Dataset 1 + Dataset 2	7600	128x128	80/20	99.65	100.00	99.30	99.60
Our Study	2023	ResNet50	Dataset 1 + Dataset 2	7600	128x128	80/20	97.21	100.00	94.00	97.00
Our Study	2023	Proposed Method	Dataset 1 + Dataset 2	7600	128x128	80/20	100.00	100.00	100.00	100.00

In the research by Kuraparthi et al. (2021), an extensive examination of brain tumor classification was conducted using three distinct CNN architectures – VGG16, AlexNet, and ResNet50 – on the Kaggle dataset. The dataset consisted of 253 images with a resolution of 224x224 pixels, and a 70/30 split was employed for training and testing, respectively. Notably, ResNet50 emerged as the most effective model, achieving an impressive accuracy of 98.28%. The study also extended its analysis to the BRATS dataset, further demonstrating the robust performance of ResNet50, which achieved an accuracy of 97.87%.

Ullah et al. (2022a) delved into a comprehensive exploration of brain tumor classification by employing various deep learning models on the Kaggle Brain Tumor Classification (MRI) dataset. Models such as ResNet101, ResNet18, InceptionResNetV2, Inceptionv3, Xception, ShuffleNet, DenseNet201, ResNet50, and MobileNetV2 were utilized on a dataset of 3264 images with varying resolutions. The study adopted an 80/20 split for training and testing, revealing noteworthy achievements. In particular, InceptionResNetV2 excelled with an accuracy of 98.91%, and Xception showcased a strong performance, achieving an accuracy of 98.37%.

Amin et al. (2020) focused on employing Support Vector Machines (SVM) for brain tumor classification across diverse datasets, including Harvard, RIDER, and Nashtar Hospital Multan. SVM exhibited high accuracies, reaching a perfect 100% in certain instances. This underscores the effectiveness of SVM in the context of brain tumor classification.

Ullah et al. (2022b) contributed to the literature with the introduction of a Tumor Resnet model for the Kaggle BTD-MRI dataset. It achieved an outstanding accuracy of 99.33%.

Additionally, Molachan et al. (2021) explored the Kaggle Brain MRI Images for Brain Tumor Detection dataset using a CNN architecture, resulting in an accuracy of 91.20%. In our study, the investigation involves the merging of Dataset 1 and Dataset 2, comprising a total of 7600 images. Various models, including XGBoost, CNN, AlexNet, DenseNet121, ResNet50, and a Majority Voting ensemble, are implemented with a 128x128 pixel image resolution. The training and testing are performed using an 80/20 split, yielding impressive accuracies across all models. Particularly, the perfect accuracy is achieved by the Majority Voting ensemble, reaching 100%. This study adds valuable insights to the field by showcasing the robust performance of different models in the challenging task of brain tumor classification.

In summary, significant differences are found between the results of our models for brain tumor detection and the results of other similar studies in the literature which used similar algorithms but worked on different datasets. Our study uses more data in model training and shows higher performance results compared to other studies' results which are given in Table 8.

Conclusion

The treatment of brain tumors varies depending on the type and severity, offering options such as chemotherapy and surgery. Benign tumors, generally non-life-threatening, may impact brain functions due

to pressure and are typically treated with surgery. Malignant tumors, posing a significant threat, require personalized treatment based on factors like condition, age, location and type. Early detection is crucial for both benign and malignant tumors, providing a better chance of successful treatment. This study specifically concentrates on the early detection of brain tumors using MRI and X-ray scans. It employs a hybrid learning system that incorporates XGBoost, CNN, DenseNet121, ResNet50, and AlexNet learning approaches, employing a Majority Voting strategy. The performance results from the five learning models, combined with the majority voting approach, exhibit promising accuracy scores. To be precise, the accuracies achieved by XGBoost, CNN, DenseNet121, ResNet50 and AlexNet are 98.98%, 99.75%, 99.65%, 97.21%, and 99.84%, respectively. The Majority Voting approach, integrating predictions from all models, demonstrated an outstanding 100.00% accuracy for brain tumor classification suggesting a reliable model. Although this study primarily focuses on detecting brain tumors as either "yes" or "no," future research aims to delve into classifying brain tumors based on their types. The outcomes of this study aim to contribute to advances in medical diagnosis and treatment strategies of brain tumors.

Acknowledgments

This work has not been financed by any institution. All authors have read and approved the final version of the manuscript.

References

Aktas, K., Ignjatovic, V., Ilic, D., Marjanovic, M., & Anbarjafari, G. (2023). Deep convolutional neural networks for detection of abnormalities in chest X-rays trained on the very large dataset. *Signal,[1] Image and Video Processing, 17*, 1035–1041. https://doi.org/10.1007/s11760-022-02482-1

Amin, J., Sharif, M., Yasmin, M., & Fernandes, S. L. (2020). A distinctive approach in brain tumor detection and classification using MRI. *Pattern Recognition Letters, 139*, 118-127. https://doi.org/10.1016/j.patrec.2020.06.012

Anantharajan, S., Gunasekaran, S., Subramanian, T., & Venkatesh, R. (2024). MRI brain tumor detection using deep learning and machine learning approaches. *Measurement: Sensors, 31*(101026). https://doi.org/10.1016/j.measen.2023.101026

Babalola, F. O., Bitirim, Y., & Toygar, Ö. (2021). Palm vein recognition through fusion of texture-based and CNN-based methods. *Signal, Image and Video Processing, 15*(3), 459-466. https://doi.org/10.1007/s11760-020-01777-6

Çelik Ertuğrul, D. (2016). FoodWiki: a mobile app examines side effects of food additives via semantic web. *Journal of Medical Systems, 40*(2), 41. https://doi.org/10.1007/s10916-016-0428-1

Çelik Ertuğrul, D., & Celik Ulusoy, D. (2022). A knowledge-based self-pre-diagnosis system to predict Covid-19 in smartphone users using personal data and observed symptoms. *Expert Systems, 39*(3), e12716. https://doi.org/10.1111/exsy.12716

Çelik Ertuğrul, D., & Elçi, A. (2020). A survey on semanticized and personalized health recommender systems. *Expert Systems, 37*(4), e12519. https://doi.org/10.1111/exsy.12519[3]

Çelik Ertugrul, D., & Ulusoy, A. H. (2019). Development of a knowledge-based medical expert system to infer supportive treatment suggestions for pediatric patients. *ETRI Journal, 41*(4), 515-527. https://doi.org/10.4218/etrij.2018-0287

Çelik Ertuğrul, D., Bitirim, Y., & Yakoub Anber, B. (2020). Decision Support System for Diagnosing Diabetic Retinopathy from Color Fundus Images. *Journal of Imaging Science and Technology*. https://doi.org/10.2352/jimagingsci.2020.64.2.020401

Çelik, D., Elçi, A., Akçiçek, R., Gökçe, B., & Hürcan, P. (2014, julho). A safety food consumption mobile system through semantic web technology. In *2014 IEEE 38th International Computer Software and Applications Conference Workshops* (pp. 348-353).[4] IEEE. https://doi.org/10.1109/COMPSACW.2014.59

Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1251-1258). https://doi.org/10.1109/CVPR.2017.124

Cifci, M. A. (2022). Segchanet: A novel model for lung cancer segmentation in ct scans. *Applied Bionics and Biomechanics, 2022*. https://doi.org/10.1155/2022/6895311

Cifci, M. A., Hussain, S., & Canatalay, P. J. (2023). Hybrid Deep Learning Approach[5] for Accurate Tumor Detection in Medical Imaging Data. *Diagnostics, 13*(6), 1025. https://doi.org/10.3390/diagnostics13061025

Cui, B., Chen, X., & Lu, Y. (2020). Semantic segmentation of remote sensing images[6] using transfer learning and deep convolutional neural network with dense connection. *IEEE Access, 8*, 116744-116755. https://www.researchgate.net/figure/The-structure-of-DenseNet-121_fig2_342368106

Ertuğrul, D. Ç., & Abdullah, S. A. (2022). A decision-making tool for early detection of breast cancer on mammographic[7] images. *Tehnički Vjesnik, 29*(5), 1528-1536. https://doi.org/10.17559/TV-20220303102353

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 770-778). https://doi.org/10.1109/CVPR.2016.90

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In[9] *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 4700-4708). https://doi.org/10.1109/CVPR.2017.243

Khan, M. M., Omee, A. S., Tazin, T., Almalki, F. A., Aljohani, M., & Algethami, H. (2022). A novel approach to predict brain cancerous tumor using transfer learning. *Computational and Mathematical Methods in Medicine, 2022*. https://doi.org/10.1155/2022/1070868

Kuraparthi, S., Reddy, M. K., Sujatha, C. N., Valiveti, H., Duggineni, C., Kollati, M., Kora, P., & V, S. (2021). Brain tumor classification of MRI images using deep convolutional neural network. *International Information and Engineering Technology Association (IIETA), 38*(4), 1171-1179. https://doi.org/10.18280/iie.380424

Li, Z., Liu, F., Yang, W., Peng, S., & Zhou, J. (2021). A survey of convolutional neural networks: Analysis, applications, and prospects. *IEEE Transactions on Neural Networks and Learning Systems*. https://doi.org/10.1109/TNNLS.2021.3060410

Liu, Q., Zhang, N., Yang, W., Wang, S., Cui, Z., Chen, X., & Chen, L. (2017). A review of image recognition with deep convolutional neural network. In *Intelligent Computing Theories and Application: 13th International Conference, ICIC 2017, Liverpool, UK, August 7-10, 2017, Proceedings, Part I 13* (pp. 69-80). Springer International Publishing. https://doi.org/10.1007/978-3-319-63307-5_8

Mikhailova, V., & Anbarjafari, G. (2022). Comparative analysis of classification algorithms on the breast cancer recurrence using machine learning. *Medical & Biological Engineering & Computing, 60*, 2589–2600. https://doi.org/10.1007/s11517-022-02685-6

Mitchell, R., Adinets, A., Rao, T., & Frank, E. (2018). Xgboost: Scalable gpu accelerated learning. *arXiv preprint arXiv:1806.11248*.

Moirangthem, M., Singh, T. R., & Singh, T. T. (2021). Image classification and retrieval framework for brain tumour detection using CNN on ROI segmented MRI images. In *2021 5th International Conference on Electrical, Electronics, Communication, Computer Technologies and Optimization Techniques (ICEECCOT)* (pp. 696-700). IEEE. https://doi.org/10.1109/ICEECCOT52177.2021.9696016

Molachan, N., Manoj, K. C., & Dhas, D. A. S. (2021). Brain Tumor Detection that uses CNN in MRI. In *2021 Asian Conference on Innovation in Technology (ASIANCON)* (pp. 1-7). IEEE. https://doi.org/10.1109/ASIANCON53535.2021.9544927

Panigrahi, A. (2021). *MRI Brain Tumor Dataset*. Kaggle. https://www.kaggle.com/datasets/abhranta/brain-tumor-detection-mri

Rawat, W., & Wang, Z. (2017). Deep convolutional neural networks16 for image classification: A comprehensive review. Neural Computation, 29(9), 2352-2449. https://doi.org/10.1162/NECO_a_00990

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., & Fei-Fei,17 L. (2015). ImageNet large scale visual recognition challenge. International Journal of Computer Vision, 115, 211-252. https://doi.org/10.1007/s11263-015-0816-y

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). MobileNetV2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 4510-4520).[18] https://doi.org/10.1109/CVPR.2018.00474

Simonyan, K., & Zisserman, A. (2014). *Very deep convolutional networks for large-scale image recognition*. arXiv preprint arXiv:1409.1556. https://doi.org/10.48550/arXiv.1409.1556

Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. (2017). Inception-v4, Inception-ResNet and the impact of residual connections on learning. In *Proceedings of the AAAI[21] Conference on Artificial Intelligence* (Vol. 31, No. 1). https://doi.org/10.1609/aaai.v31i1.11239

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the Inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2818-2826). https://doi.org/10.1109/CVPR.2016.308

Tazin, T., Sarker, S., Gupta, P., Ayaz, F. I., Islam, S., Monirujjaman Khan, M., Alroobaea, R., & Alshazly, H. (2021). A robust and novel approach for brain tumor classification using convolutional neural network. *Computational Intelligence and Neuroscience, 2021*. https://doi.org/10.1155/2021/6638068[22]

Ullah, N., Khan, J. A., Khan, M. S., Khan, W., Hassan, I., Obayya, M., Qamar, T., & Salama, A. S. (2022a). An effective approach to detect and identify brain tumors using transfer learning. *Applied Sciences, 12*(11), 5645. https://doi.org/10.3390/app12115645

Ullah, N., Khan, M. S., Khan, J. A., Choi, A., & Anwar, M. S. (2022b). A robust end-to-end deep learning-based approach for effective and reliable BTD using MR images. *Sensors, 22*(19), 7575. https://doi.org/10.3390/s22197575

Uyguroğlu, F., Toygar, Ö., & Demirel, H. (2024). CNN-based Alzheimer's disease classification using fusion of multiple 3D angular orientations. *Signal, Image and Video Processing, 18*, 2743–2751. https://doi.org/10.1007/s11760-023-02685-z

Üzülmez, S., & Çifçi, M. A. (2024). Early diagnosis of lung cancer using deep learning and uncertainty measures. *Journal of the Faculty of Engineering and Architecture of Gazi University, 39*(1), 385-400. https://doi.org/10.17341/gazimmfd.1215701

Viradiya, P. (2021). *X-Ray Brain Tumor Dataset*. Kaggle. https://www.kaggle.com/preetviradiya/brian-tumor-dataset

Wang, W., Chakraborty, G., & Chakraborty, B. (2020). Predicting the risk of chronic kidney disease (ckd) using machine learning algorithm. *Applied Sciences, 11*(1), 202. https://doi.org/10.3390/app11010202

Zhang, X., Zhou, X., Lin, M., & Sun, J. (2018). ShuffleNet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 6848-6856). https://doi.org/10.1109/CVPR.2018.00715