

Bayesian Optimization-based CNN Framework for Automated Detection of Brain Tumors

Mahir Kaya


Abstract— Brain tumors, capable of yielding fatal outcomes, can now be identified through MRI images. However, their heterogeneous nature introduces challenges and time-consuming aspects to manual detection. This study aims to design the optimal architecture, leveraging Convolutional Neural Networks (CNNs), for the automated categorization of brain tumor types within medical images. CNN architectures frequently face challenges of overfitting during the training phase, mainly attributed to the dual complexities of limited labeled datasets and complex models within the medical domain. The depth and width hyperparameters in these architectures perform a vital role, in determining the extent of learning parameters engaged in the learning process. These parameters, encompassing filter weights, fundamentally shape the performance of the model. In this context, it is quite difficult to manually determine the optimum depth and width hyperparameters due to many combinations. With Bayesian optimization and Gaussian process, we identified models with optimum architecture from hyperparameter combinations. We performed the training process with two different datasets. With the test data of dataset 1, we reached 98.01% accuracy and 98% F1 score values. With the test data of dataset 2, which has more data, 99.62% accuracy and F1 score values were obtained. The models we have derived will prove valuable to clinicians for the purpose of brain tumor detection.

Index Terms— Deep learning, CNN, Bayesian optimization, Brain tumor.

I. INTRODUCTION

BRAIN TUMORS are fatal, and life span of patients can be quite short [1]. Magnetic resonance imaging (MRI) is the most well-known and successful tool for detecting and classifying brain tumors due to its aptitude for distinguishing between structure and tissue depending on contrast levels [2]. The detection and classification of brain tumor is often determined manually by the clinician; the duration of this process is quite long and sometimes can lead to erroneous results [3]. Early detection of such tumors can improve the effectiveness of the therapeutic process and increases the likelihood of long-term survival [4]. The inherent heterogeneity within brain tumor cells poses a hurdle in precisely classifying the tumor type, consequently hindering treatment planning.

MAHİR KAYA, is with Department of Computer Engineering, Faculty of Engineering and Architecture, Tokat Gaziosmanpaşa University, Tokat, Türkiye. (e-mail: mahir.kaya@gop.edu.tr)

 <https://orcid.org/0000-0001-9182-271X>

Manuscript received Jan 16, 2023; accepted October 4, 2023.

DOI: [10.17694/bajece.1346818](https://doi.org/10.17694/bajece.1346818)

The principal objective of this study is to correctly identify brain tumor types, thereby facilitating informed decisions regarding suitable treatment approaches.

Artificial intelligence applications are used in many areas such as cloud computing [5, 6] and computer-assisted disease diagnosis [7]. Convolutional Neural Networks (CNNs), which are a part of deep learning, demonstrate exceptional performance in end-to-end learning. During the training phase, they autonomously generate feature maps from input images and subsequently execute classification in the network's final stages. The network's parameters undergo updates based on error rates during the concluding phase [8]. One of the important contrasts between machine learning and deep learning models hinges upon the choice of features employed for classification. While traditional machine learning relies on manually crafted features, deep learning directly uncovers features from data, eliminating the need for external intervention [9, 10]. As a result, CNN models require an ample supply of images to achieve successful training outcomes. However, within the domain of healthcare, there's generally a shortage of labeled data. This shortage, combined with the intricate nature of CNN models, often leads to the occurrence of memorization issues during the training phase [11]. Identifying the optimal architecture (comprising depth and width) along with a suitable hyperparameter combination is a crucial challenge when aiming for successful learning within CNN models constrained by a limited dataset.

CNN architectures are typically composed of sequential convolutional, max-pooling, and fully connected layers. In the Convolution layer, feature maps are generated by applying various filters to the raw image. This process is iterated across subsequent convolutional layers. In later layers, a higher number of filters is commonly employed to yield more intricate feature maps. The max-pooling layer reduces the feature map dimensions. These two layers collaboratively conduct the essential task of feature extraction from raw images within CNN architectures [12]. During the training phase, the filter weights—located where the learning process occurs—are updated using the backpropagation algorithm at the outcome layer, based on the error rate [8]. These filter weights are initially set randomly. The fully connected layer is often designated as the classification layer. Depending on the class count, the last layer incorporates either sigmoid or softmax functions. Of paramount importance is the determination of the optimal count of convolutional layers, along with the quantity

and size of filters within each convolutional layer, to foster successful learning in CNN architectures.

Increasing the model's depth and width typically leads to two prevalent challenges. One of these issues pertains to the vanishing gradient problem, while the other revolves around the risk of overfitting the training dataset. The vanishing gradient problem can be mitigated through the integration of residual connections. Nonetheless, even when employing generalization techniques such as batch normalization, data augmentation, and dropout, identifying the point at which excessive learning initiates remains crucial. This can be discerned by monitoring loss/accuracy graphs during the training phase. In shallower models with reduced model depth and width, learning is hindered, resulting in underfitting and lower performance. A major concern to tackle is the process of identifying the most suitable depth and width hyperparameters within CNN models. Given the often extensive training period these models require, attempting all conceivable combinations to obtain results can be a time-intensive process. Addressing this, Bayesian optimization in conjunction with Gaussian processes offers an effective means to efficiently pinpoint the optimal hyperparameter combinations. Within this study, we identified the optimal CNN architecture to achieve accurate brain tumor type detection using two distinct datasets. Employing Bayesian optimization, we systematically arrived at optimal hyperparameter values by iteratively combining them with model accuracy data. The resulting models hold promise in aiding clinicians with brain tumor type detection.

The contributions of this study can be listed as follows:

- A novel and effective CNN model has been developed to detect brain tumor types.
- This study presents an analysis of the efficacy of different CNN architectures, encompassing a range of widths and depths, in the context of brain tumor type detection.
- The optimal hyperparameter combinations were identified using Bayesian optimization in conjunction with Gaussian processes.
- In the context of two separate datasets, it's notable that the models' performance sees improvements in instances characterized by a larger data volume.

The structure of this work is as follows: In the Related Works section, we provide a summary of previous research on brain tumor detection and classification. Section 3 delves into the specifics of the dataset features, the Bayesian optimization method, and the overall structure of the CNN model. Section 4 presents the outcomes of our proposed models and compares them with findings from other research. In the conclusion, we underscore the significance of this study, its broader contributions, and potential paths for future research.

II. RELATED WORKS

Brain MR images are frequently used in research in a variety of fields, including tumor segmentation and tumor type

classification. The research on brain tumor classification can be divided into three categories: tumor detection, tumor type classification, and tumor detection and classification.

The first of these is the studies carried out to detect the presence of tumor in brain MR images. The datasets contain two classes, tumor and normal. Toğaçar et al. [13] proposed a model for identifying brain tumors and the study used a dataset of 253 MRI images. The proposed model employs the attention module as well as the hypercolumn technique. The attention module is used to detect important areas of the image, and the hypercolumn technique provides more effective feature selection by allowing data from each layer to be used in the final layer, according to the study. The proposed model reached a 96.05 percent accuracy in brain tumor detection. Using the data augmentation technique, the authors [14] augmented the unbalanced dataset (155 tumor, 98 normal) to include an equal number of data for the tumor (155) and normal (155) classes. In the study, features obtained by hypercolumn technique using pre-trained AlexNet and VGG16 architectures were reduced by the recursive feature elimination (RFE) method and classified by SVM. The models that were classified using 200 and 300 features yielded the best accuracy values in the study, 96.77%. Balamurugan and Gnanamanoharan [15] proposed a hybrid CNN model for the detection of brain tumors. The dataset used in the study comprises 271 tumor images and 98 non-tumor images. There are 173 images set aside for training, 50 for validation, and 30 for testing. A Laplacian Gaussian filter (LOG) was used for data preprocessing in the study, and a Fuzzy C Means with Gaussian mixture model (FCM-GMM) algorithm was used for brain tumor segmentation. 13 features were determined using the VGG-16 architecture as a feature extractor, and classification was performed using the proposed enhanced LuNET algorithm. The proposed model's performance metrics are as follows: accuracy (99.7%), sensitivity (98.2%), specificity (98.6%), precision (99.4), F-Score (98.2), and recall (99.8%).

The second category includes studies on brain tumor classification. The researcher employs various datasets containing three or four types of brain tumors. Deepak and Ameer [16] undertook a study with the objective of classifying three different brain tumor varieties. In the study, GoogleNet architecture was used with transfer learning method for the purpose of feature extraction from brain MRIs. In addition to softmax as a classifier, SVM and KNN algorithms have also been tried. In the study, the best accuracy rate of 98% was obtained with the KNN algorithm and 80% of the dataset was used as the training set. The model was trained using 70%, 50% and 25% of the dataset, and classifications were performed with SVM. It has been reported that shrinking the training dataset does not significantly affect performance. To classify brain tumors, Başaran [17] proposed a hybrid model. Gray level co-occurrence matrix (GLCM) and Local Binary Pattern (LBP) algorithms, as well as four CNN models: AlexNet, VGG16, EfficientNetB0, and ResNet50, were used to obtain the features. Artificial Bee Colony (ABC), Particle Swarm Optimization (PSO), and Genetic algorithms (GA) were implemented to reduce these properties, and SVM was used to classify them.

The highest accuracy obtained in the study was 98.22% with 5-fold cross validation and an 86%/14% training/test ratio. Ayadi et al. [18] presented a model designed for brain tumor classification and this model was tested using three datasets. The proposed CNN model comprises of 10 convolutional layers, a dense layer, and a classifier based on softmax. The study's best performance was obtained with the Adagrad optimization algorithm and 0.003 learning rate, training for 20 epochs, and batch size 16. The model was trained and tested using the Figshare dataset in the study. The tests were then repeated with two different datasets (Radiopaedia and REMBRANDT) to provide additional validation. The dataset accuracy values are as follows: dataset1 (94.74%), dataset2 (93.71%), and dataset3 (five sub-datasets: 100%, 97.22%, 97.02%, 88.86%, and 95.72%). Ait-Amou et al. [19] use Bayesian optimization to determine the proposed CNN model's hyperparameters. The study involved an examination of the classification of three brain tumor types, namely glioma, meningioma, and pituitary. Their base model includes 5 convolutional blocks, a dense block, and a classification block. The input size is 64*64 pixels and softmax activation function was used for classification. Bayesian optimization was employed to identify the optimal dropout rate, number of dense nodes, activation function, batch size, and optimization algorithm that yielded the best results in the study. The best performance, according to the results, was 98.70%. Alhassan and Zainon [20] explored the effect of the activation function on brain tumor classification. Feature extraction from brain images was accomplished using the Histogram of Oriented Gradients (HOG) technique, and a classification model employing the Hard Swish-based ReLU activation function was introduced for brain tumor classification. Assessment of the suggested model's effectiveness was conducted using two different architectures (CNN and RNN), a selection of three activation functions (sigmoid, tanh, and Hard Swish-based ReLU), and four varied training and testing ratios (20:80, 40:60, 60:40, 80:20). The top accuracy score was obtained by the CNN model that utilized the Hard Swish-based ReLU activation function, in conjunction with an 80:20 ratio. The approach presented by Aurna et al. [21] involves a two-stage process for feature selection in brain tumor classification. A total of four datasets were used in the study, including three individual datasets and one obtained by combining them. The best feature extractors were identified among the set of five pre-trained models and a new model (Scratched CNN). In the first stage, the three best models (EfficientNet-B0, ResNet-50, and Scratched CNN) were combined in pairs, and the model pairs with the highest accuracy value were determined. In the second stage, features were acquired using these pairs of models. During the feature reduction phase, Principal Component Analysis was used. For the classification process, the performance of Softmax, SVM, RF, KNN, and AdaBoost algorithms was compared, and Softmax produced the best results. The proposed model for the combined dataset produced the best accuracy performance of 98.96%. Accuracy values for other data sets are 99.67 for dataset1, 98.16 for dataset2, and 99.76 for dataset3, respectively. Mehnatkesh et al. [10]

conducted a hyperparameter optimization study to classify brain tumors. First, the images' empty space was cropped, and data augmentation was implemented. The performance of seven different state-of-the-art models was then analyzed on the dataset, and the ResNet model with the best result was chosen. The optimization study yielded 99.02% classification success with the ResNet model and the Improved Ant Colony Optimization algorithm. Kazemi et al. [22] used the Figshare and TCIA datasets in their research to determine the classification of brain tumors in MRI scans. In the proposed model, AlexNet and VGG16 architectures are trained concurrently. SVM, KNN, and Decision Tree methods were used to choose the most essential characteristics derived from the two architectures. The Softmax classifier was used to predict the tumor class. They investigated the suggested model's performance in binary and multiclass classification. The best accuracy result with Figshare dataset is 99.14% in binary class and 98.78% in multi-class. Gomez-Guzman et al. [23] compared the performance of seven different CNN models for brain tumor classification. A 17-layer CNN model with four convolutional layers was proposed. In addition, six pre-trained architectures: ResNet50, MobileNetV2, Xception, InceptionV3, InceptionRes-NetV3, and EfficientNetB0 were used in the study. A dataset of 7023 MRI images with four classes: no-tumor, glioma, meningioma, and pituitary was used in the study. The best accuracy was 97.12% with the InceptionV3 model. Türkoğlu[24] proposed a four-stage hybrid system for brain tumor diagnosis. The images are enhanced by preprocessing first. Transfer learning method was then applied to DenseNet and AlexNet architectures and 4096 and 1000 deep features were obtained respectively. The most significant features were determined in the third stage through feature reduction using the MrMr algorithm on the combined features. The SVM algorithm was employed in the final stage to ascertain the class to which the tumors belong. The Bayesian Optimization Algorithm was implemented to optimize the SVM classifier's hyperparameters. The author used the figshare dataset, which included 3064 brain MRI images and three classes of brain tumors. The proposed model was tested by selecting eight different numbers (from 500 to 5000) of combined deep features and the best accuracy performance was 98.04% using 2500 features. In addition, the author conducted an extensive experimental study on feature extraction from CNN models, feature selection (MrMr), and the optimization of SVM hyperparameters using Bayesian optimization for classification. The features obtained from the CNN models are reduced by the MrMr method and then used as input for the machine learning model. The difference in our work is that with the Gaussian process-based Bayesian optimization algorithm, many hyperparameters in the CNN architecture such as the number of convolution layers, number and size of filters, learning rate, dropout rate, and optimizer were optimized. We optimized directly on the CNN architecture without the need for features extracted from CNN models and a two-stage training process.

In the final category, there are studies that both detect and classify brain tumors. Mondal and Shrivastava [25] conducted

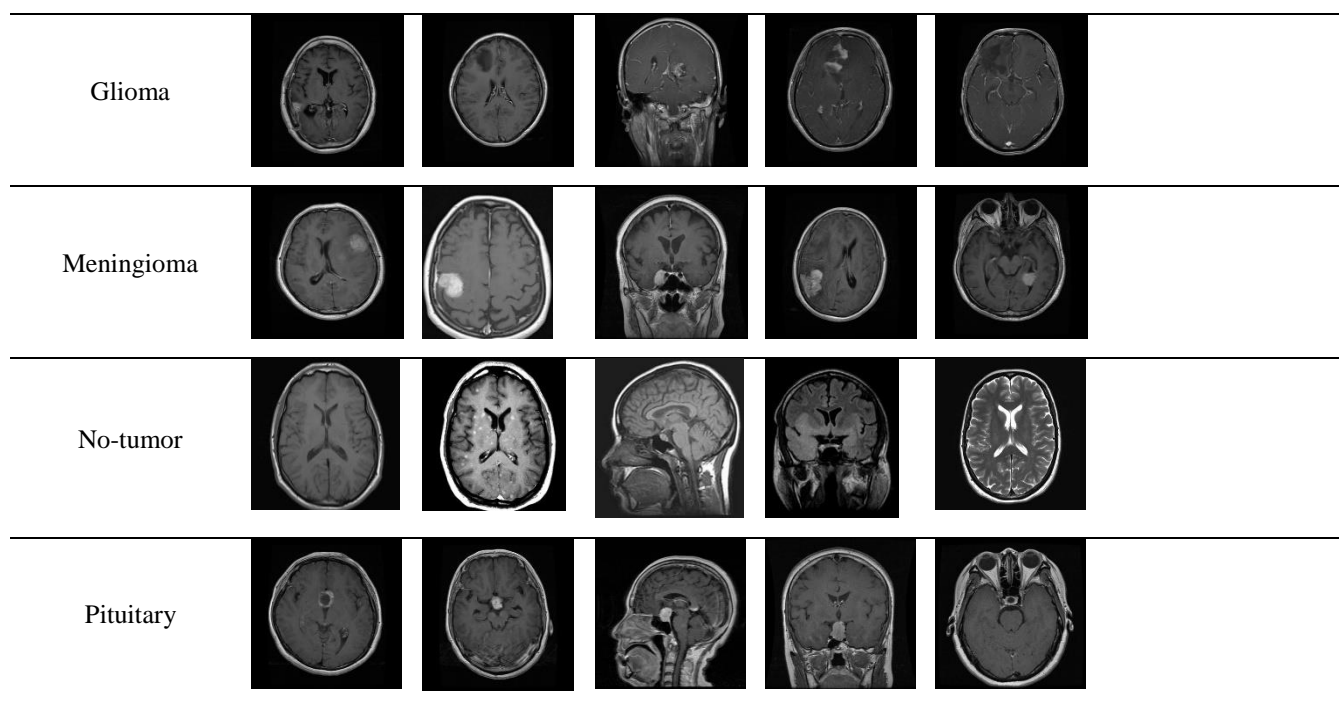


Fig. 1. Examples of three types of tumors and normal brain MR images

a study involving two separate datasets to identify and categorize brain tumors. They introduced a novel activation function named Parametric Flatten-p Mish (PFpM) alongside the BMRI-Net model. The model accomplished an accuracy of 99.00% in the detection of brain tumors and 99.57% in the classification of tumor type. Saurav et al. [26] devised a CNN model that employs channel-attention blocks to concentrate on pertinent areas within the image when classifying tumors. The selection of the pertinent feature maps is carried out via channel-attention blocks. The suggested model's performance was evaluated using four different datasets. The BT-small-2c and BT-large-2c datasets were utilized for tumor detection, and the other two were used for tumor categorization (BT-large-3c and BT-large-4c). According to the datasets, the following accuracy values were obtained: BTsmall-2c (96.08%), BT-large-2c (99.83%), BT-large-3c (97.23%), and BT-large-4c (95.71%). Turk et al. [27] proposed a system that detects and classifies brain tumors. The study was performed in three stages. First, brain tumor detection was performed with 2 classes (tumor and normal), then brain tumor classification was performed with 4 classes (glioma, meningioma, pituitary and normal) and finally Class Activation Maps were created. In the study, 3441 MR images for the first stage and 3362 MR images for the second stage from two different datasets were used. The transfer learning method was performed with the ensemble DL approach using ResNet50, VGG19, InceptionV3 and MobileNet architectures. The highest accuracy rate was 100% for brain tumor detection (with InceptionV3, MobileNet and ResNet50 architectures) and 96.45% for tumor classification (with ResNet50 architecture). Alanazi et al. [28] designed three scratch CNN models with 19, 22 and 25 layers to detect brain tumor and compared their performance. They obtained the best accuracy of 92.67% with the 22-layer CNN model. Then, they trained a model that detects the type of brain tumor using the

22-layer CNN model with a fine-tuning approach using the transfer learning. The test accuracy of this model is 95.75%. Kang et al. [29] proposed a feature ensemble-based model for brain tumor classification and investigated the performance of nine different ML classifiers. They used 13 different CNN architectures to extract features to be used in classification. The three models that provided the best features were determined using ML classifiers. DenseNet-169, ShuffleNet V2, and MnasNet provided the best features for four-class (glioma, meningioma, pituitary and normal) classification. The features from these three models were combined and fed into ML classifiers to identify brain tumor classes. The best accuracy for brain tumor diagnosis was 93.72% with SVM (RBF) for four-class classification and 98.83% for two-class classification.

Numerous studies in the literature have explored brain tumor detection. Transfer learning methods have been employed, typically yielding successful results in scenarios with limited datasets. However, current state-of-the-art CNN models tend to be intricate, as they are primarily tailored to vast datasets like ImageNet. In the context of medical images with limited labeled data [30], these models often grapple with issues such as overfitting or memorization during training. Despite the application of techniques like batch normalization, data augmentation, L2 regularization, and dropout to mitigate these challenges, they often fall short of being entirely effective. While the approach of utilizing CNN models for feature extraction followed by machine learning classification has become prevalent, it necessitates a two-stage training process

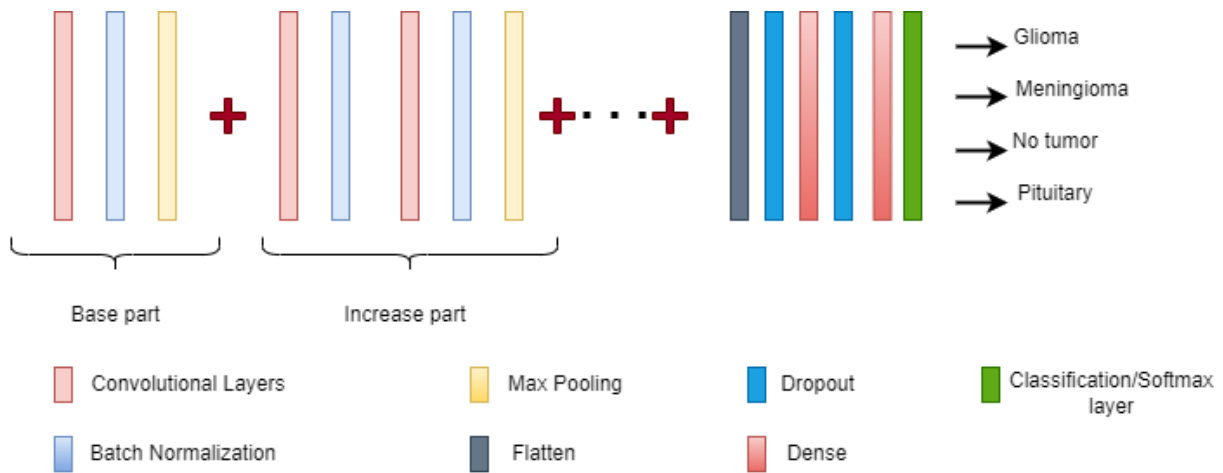


Fig. 2. Bayesian optimization-based proposed model structure

and can be demanding to implement more complexity. The key to success often lies in a well-designed CNN architecture, particularly in optimizing hyperparameters such as depth and width to suit the specific dataset. In this study, we addressed this challenge by determining the optimal depth and width hyperparameters for CNN architectures using Bayesian optimization.

III. MATERIALS AND METHOD

A. Dataset

Two datasets were used in this study. Dataset-1 [31], a publicly accessible brain tumor dataset, has a total of 3264 brain MRIs. The dataset-1 has four classes: glioma, meningioma, pituitary and healthy (no tumor). This dataset contains 926 gliomas, 937 meningiomas, 901 pituitary, and 500 healthy images. Dataset-2 [32] represents a publicly available dataset containing brain tumors and has a total of 7023 brain MRIs. This dataset was created by merging three separate datasets (Figshare, SARTAJ, and Br35H). The dataset consists of four distinct classes. These are MR images of healthy, meningioma, pituitary, and glioma patients' brains. There are 2000 images of healthy people, 1621 gliomas, 1645 meningiomas, and 1757 pituitary tumors. The dataset divisions for training, validation, and testing are presented in Table 1. Fig. 1 shows examples of three types of tumors and normal brain MRI images from dataset 2.

TABLE I
TRAIN, VALIDATION AND TEST PART OF DATASETS

	Train	Validation	Test
Dataset 1¹	2351	261	652
Dataset 2²	5141	571	1311

¹Sartaj dataset <https://www.kaggle.com/sartajbhuvaji/brain-tumor-classification-mri>

²Massoud dataset <https://www.kaggle.com/datasets/masoudnickparvar/brain-tumor-mri-dataset?select=Training>

B. Bayesian optimization

The optimization approach involves iteratively generating neural networks with defined hyperparameters, executing the

training process, and determining the optimum hyperparameter set among these constructed networks. The Sequential model-based optimization (SMBO) approach, that is Bayesian optimization, was utilized in this study to estimate hyperparameters and network design [33, 34].

Bayesian optimization is a technique based on probabilistic modeling, used to find the peaks or lowest points of objective functions that are costly to assess. It can be used when the goal function has no closed-form expression but observations of this function can be acquired at sampling values [35, 36]. An acquisition function is used within the context of Bayesian optimization to effectively select the next sampling location. It automates the balance between exploration and exploitation. Exploration takes place when there's uncertainty in the objective function, whereas exploitation centers on utilizing x values where the objective function is anticipated to be at its peak [35]. The primary goal of this optimization method is to reduce the count of evaluations for the objective function, making it advantageous. The Bayes theorem is employed to compute the posterior probability of an event, taking into account both the prior probability and the likelihood probability of the event. In cases where the objective function is uncertain, the Bayesian model provides an elegant approach to defining attributes of the objective function with the aid of informative priors, such as approximate locations of the maximum or its smoothness [33, 36].

In this research, we utilized the Gaussian Process (GP) method and gathered the initial $D_i(x_i, y_i)$ data. Through the input X , we conducted training on the dataset to acquire the function output y . At any given point x , the value of f_x is treated as a stochastic variable. The random variables f_{x_i} and f_{x_j} , corresponding to distinct x_i and x_j points, exhibit correlation. To represent these random variables, we employed a Gaussian

distribution ($f(x) \sim N(\mu(x), \sigma^2)$). Given that we conducted optimization for several hyperparameters, we employed the Gaussian Process (GP) as presented in Equation 1. The fundamental aspect of a GP is its function distribution, and this is completely characterized by the mean and covariance functions. The kernel function in Equation 2 was applied to a

set of hyperparameters with m dimensions. It quantifies the similarity between two distinct predictions. We employed the maximum likelihood estimate (MLE) approach, as described in Equation 3 [37], for hyperparameter selection. In this method, we evaluate the likelihood of the observations $f(x_{1:n})$ with respect to the prior distribution, $P(f(x_{1:n})|\varphi)$, which follows a multivariate normal density. Here, φ represents the hyperparameter vector. Afterward, we estimate φ using the maximum a posteriori (MAP) estimation, which corresponds to the value of φ that maximizes the posterior distribution [37].

$$f_x \sim GP(m(x), k(x, x')) \quad (1)$$

$$k(x_i, x_j) = \exp\left(-\frac{1}{2}\|x_i - x_j\|^2\right) \quad (2)$$

$$\begin{aligned} \varphi^* &= \operatorname{argmax}_{\varphi} P(\varphi|f(x_{1:n})) \\ &= \operatorname{argmax}_{\varphi} P(f(x_{1:n})|\varphi)P(\varphi) \end{aligned} \quad (3)$$

$$\begin{aligned} EI(x) &= \begin{cases} (\mu(x) - f(x^*))\Phi(Z) + \sigma(x)\phi(Z) & \text{if } \sigma(x) > 0 \\ 0 & \text{if } \sigma(x) = 0 \end{cases} \end{aligned} \quad (4)$$

$$Z = \frac{\mu(x) - f(x^*)}{\sigma(x)}$$

Within Bayesian optimization, a critical step revolves around the criteria used to determine the upcoming collection of hyperparameters derived from the surrogate function. The Expected Improvement, provided in Equation 4, stands out as the most commonly adopted criterion [35]. By considering the expected value of the improvement function concerning the Gaussian process's predictive distribution, we achieve a harmonious equilibrium between exploration and exploitation. During exploration, our emphasis is on pinpointing locations characterized by substantial surrogate variance. Conversely, during exploitation, we direct our attention to points with elevated surrogate means [37].

C. Proposed Method

In this study, we initially identified a fundamental block comprising convolutional, batch normalization, and max pooling layers. Following this foundational block, we introduced a max pooling layer after every 2 convolutional and batch normalization layers and elucidated the growth block structure for each model proposal. Fig. 2 shows the general structure of the proposed method.

Following activation functions in the convolutional layers, the batch normalization layer aids in achieving quicker convergence towards optimal values for the models, while also preventing overfitting during the training phase. In deep learning models, as the number of convolutional layers increases—leading to greater depth—it is anticipated that more intricate feature maps will be acquired. In light of this, we devised a block structure that incorporates max pooling after every two successive convolutional layers. By amplifying the count of these blocks, we effectively enhance the depth within the models. Increasing the depth of a model often gives rise to two primary challenges. The first involves the vanishing gradient, while the second pertains to the potential overfitting

of the training dataset. The vanishing gradient dilemma can be alleviated via the use of residual connections. However, even with generalization techniques like batch normalization in place, recognizing the juncture at which undue learning takes hold remains pivotal. This can be deduced by closely observing the loss/accuracy graphs during the training phase. In this study, we employ Bayesian optimization to identify the most suitable model depth-width and optimal hyperparameters from a range of possibilities.

IV. EXPERIMENT AND RESULTS

CNN architectures automate the extraction and classification of features directly from input images, bypassing the need for manual feature extraction as required by traditional machine learning algorithms. Within CNN architectures, the training process involves the movement of each image through the network, and learning takes place as filter weights are updated using the backpropagation algorithm based on error rates at the output layer. Alongside this, various hyperparameters are defined for each training stage.

The determination of optimal hyperparameters, which maximize classification performance, stands as a significant concern. Hyperparameter value ranges are presented in Table 2. These ranges were established for the brain tumors dataset following numerous trial-and-error iterations.

TABLE II
HYPERPARAMETERS AND VALUES

Hyperparameters	Values
Convolutional layer size	5, 7, 9, 11
Kernel size	3x3, 5x5
Filters size	min value :=16, max value :=256, step :=16
Dropout-rate	0.0, 0.2, 0.3, 0.4, 0.5, 0.6
Optimizer	Adam, SGD with Nesterov
Learning rate	0.001, 0.0001

CNN architectures commonly achieve favorable outcomes through the implementation of deep networks, allowing for detailed feature extraction. Nevertheless, in cases where the available labeled data is scarce, the training phase frequently encounters challenges related to memorization, thereby detrimentally impacting overall performance. Consequently, despite the models exhibiting high training success, their performance on unseen test datasets remains notably suboptimal.

As can be seen in Table 3, models with four different depths and filter numbers are reported on dataset 1. Out of the various configurations, the 9-conv-layer CNN architecture coupled with the hyperparameter settings in Model 3, determined through Bayesian optimization, yielded the most optimal results for dataset 1. When working with a constrained dataset size, elevating the model's depth can result in reduced performance on the test dataset, primarily due to the risk of memorization during the training phase. In Dataset 1, Model 3 exhibited the

most impressive performance, achieving a test accuracy of 98.01%.

TABLE III
MODEL STRUCTURES BASED ON BAYESIAN OPTIMIZATION FOR DATASET 1

Layers	CNN Model 1	CNN Model 2	CNN Model 3	CNN Model 4
Conv-1	3x3, 16	3x3, 48	3x3, 112	3x3, 112
Conv-2	5x5, 16	3x3, 48	5x5, 112	3x3, 112
Conv-3	3x3, 176	5x5, 128	3x3, 112	3x3, 112
Conv-4	5x5, 256	3x3, 80	5x5, 112	5x5, 112
Conv-5	5x5, 208	3x3, 128	3x3, 240	3x3, 240
Conv-6	-	5x5, 256	3x3, 240	5x5, 240
Conv-7	-	5x5, 48	5x5, 240	3x3, 16
Conv-8	-	-	5x5, 144	5x5, 16
Conv-9	-	-	5x5, 48	5x5, 112
Conv-10	-	-	-	5x5, 48
Conv-11	-	-	-	3x3, 16
Dropout-rate-1	0,6	0,6	0,4	0
Dense-1	256	256	208	64
Dropout-rate-2	0	0	0	0
Dense-2	256	256	64	256
Optimizer	SGD with Nesterov	SGD with Nesterov	SGD with Nesterov	SGD with Nesterov
Learning-rate	0.001	0.0001	0.0001	0.001
Accuracy Score (%)	95.40	96.47	98.01	96.78

axa,b : a stands for kernel size and b stands for filter size in convolutional layers

The optimal model configuration for dataset 2 is illustrated in Fig. 3. In this dataset, the 9-conv-layer convolutional architecture demonstrated exceptional performance, achieving a test accuracy of 99.62%. Beyond Bayesian optimization, the notable success of this architecture can be attributed to the substantial increase in the volume of data. In CNN architectures, it's generally expected that the number of filters will be higher in the later layers to facilitate more intricate feature extraction. An intriguing observation we made using Bayesian optimization in both datasets is the decline in filter counts in the last layers. This reduction can be attributed to Bayesian optimization addressing instances of overfitting during the training phase, often associated with excessive filter numbers.

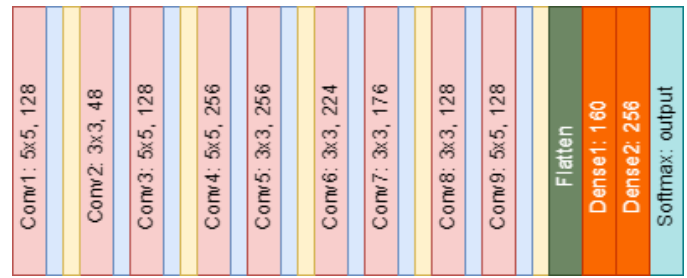
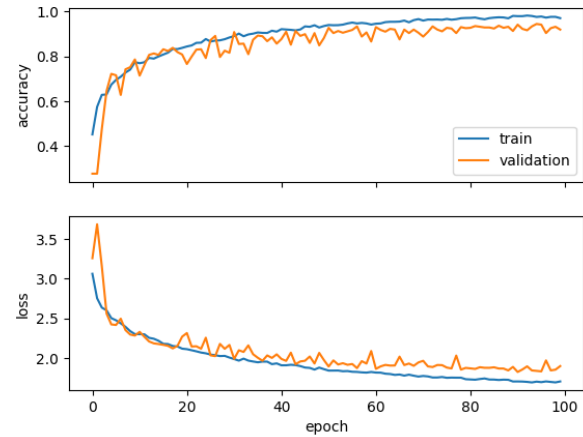
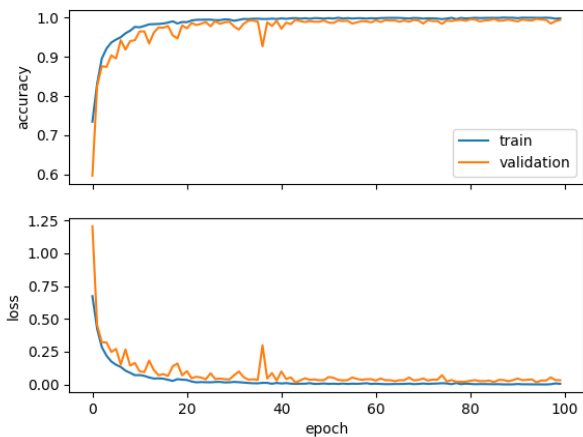


Fig. 3. Optimum model structure for dataset 2



(a)



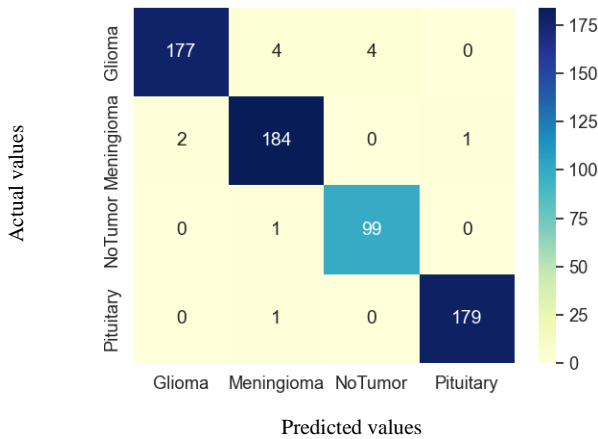
(b)

Fig. 4. Training and Validation accuracy/loss values for (a) dataset 1 (b) dataset 2

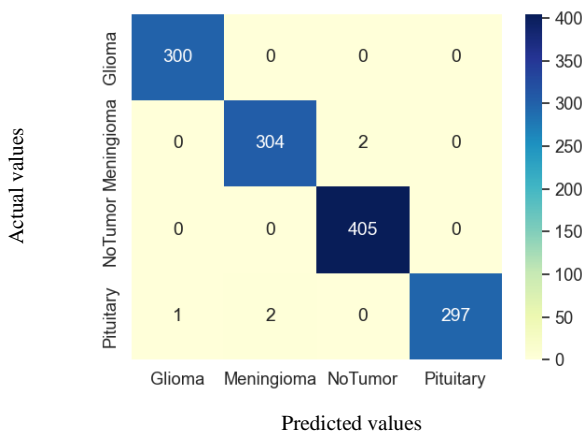
Fig. 4 depicts the training and validation loss/accuracy plots for both dataset 1 and dataset 2. From these graphs, it's evident that the models achieving optimal outcomes do not exhibit signs of memorization during the training phase. In cases of memorization, the training accuracy consistently improves while the validation accuracy tends to decline after reaching a certain epoch value. Similarly, during memorization, the training loss demonstrates a continuous decline in the loss graph, while the validation loss tends to rise after a specific epoch value. Across both graphs, a consistent trend of increase/decrease is observed in the training and loss graphs,

with instances of overlap at various points. Upon examining Fig. 4b, it becomes evident that due to the increased volume of data in dataset 2, the training and validation curves exhibit a noticeable overlap during the training phase.

Additionally, within the Meningioma column, 2 Pituitary images were erroneously labeled as Meningioma.



(a)



(b)

Fig. 5. Confusion matrix of (a) dataset 1 and (b) dataset 2

Fig. 5 presents the confusion matrix results generated by the best models on dataset 1 and dataset 2. Upon inspection of Figure 5a, it becomes evident that out of the total 185 glioma disease images, 177 were accurately predicted. Consequently, the True Positive count stands at 177. Furthermore, there were instances where 4 images, originally categorized as gliomas, were wrongly identified as meningiomas, while another 4 were classified as No tumor. These misclassifications amount to a total of 8 images, which constitute the False Negative instances. Moreover, examining the column corresponding to glioma, we find that 2 meningioma images were erroneously labeled as gliomas. These 2 instances contribute to the False Positive values within the column. Upon reviewing Fig. 5b, it's evident that all glioma images were accurately classified. Furthermore, upon closer inspection of the Meningioma category, it was revealed that 2 images that were supposed to be classified as meningioma were instead wrongly predicted as No tumor.

TABLE IV
PERFORMANCE METRICS OF PROPOSED MODEL FOR DATASET 1

Classes	Precision	Recall	F1-Score	ICSI	Support
Glioma	0.9888	0.9568	0.9725	0.9456	185
Meningioma	0.9684	0.9840	0.9761	0.9524	187
No-tumor	0.9612	0.99	0.9754	0.9512	100
Pituitary	0.9944	0.9944	0.9944	0.9888	180
Accuracy	0.9801				652
Macro avg	0.9782	0.9813	0.9796		652
Weighted avg	0.9803	0.9801	0.98		652

TABLE V
PERFORMANCE METRICS OF PROPOSED MODEL FOR DATASET 2

Classes	Precision	Recall	F1-Score	ICSI	Support
Glioma	0.9967	1.00	0.9983	0.9967	300
Meningioma	0.9935	0.9935	0.9935	0.987	306
No-tumor	0.9951	1.00	0.9975	0.9951	405
Pituitary	1.00	0.99	0.995	0.99	300
Accuracy	0.9962				1311
Macro avg	0.9963	0.9959	0.9961		1311
Weighted avg	0.9962	0.9962	0.9962		1311

TABLE VI
PERFORMANCE METRICS OF PROPOSED MODELS FOR DATASET 1 AND DATASET 2

	MCC	Kappa	CSI
Proposed Model (Dataset 1)	0.9731	0.9730	0.9595
Proposed Model (Dataset 2)	0.9949	0.9949	0.9922

Performance metric results, derived from the confusion matrix data, are presented in Table 4, Table 5 and Table 6. Within medical image-based diagnostic systems, the computation of precision and recall values holds paramount importance. Precision denotes the proportion of True Positives to all Positives (TP/(TP+FP)), whereas recall measures the model's accurate recognition of True Positives (TP/(TP+FN)). The F1 score, representing the harmonic average of precision and recall, offers a comprehensive evaluation. It's worth noting that precision and recall values can exhibit disparities, particularly in scenarios of class imbalances within the dataset.

The performance metrics for each class in dataset 1 and dataset 2 are given in Table 4 and Table 5. The precision, recall, and F1-score values in Table 4 and Table 5 were calculated as described in the reference [38]. In addition, it was also verified with the scikit-learn library. Machine learning makes use of the Matthews Correlation Coefficient (MCC) as a statistical gauge to assess the accuracy of binary and multiclass classifications. The MCC assesses the model's overall performance by taking into account not just the accuracy of predictions but also the possibility of random agreement, which is particularly relevant in scenarios involving multiple classes. A higher MCC score signifies superior model performance across all classes, whereas a lower score indicates a weaker alignment between predictions and actual labels. When it comes to multiclass classification, the Kappa score enhances the evaluation of classification performance by taking chance agreement into consideration, making it especially advantageous in scenarios where classification models need assessment and there are multiple classes or imbalances in the dataset. In the context of classification assessment, the Individual Classification Success Index (ICSI) serves as a class-specific symmetric metric. Averaging the Individual Classification Success Index (ICSI) scores for all individual classes yields the Classification Success Index (CSI), offering a holistic assessment of the classification performance. A CSI value close to 1 indicates that the classification performance is very good [39]. MCC, Kappa, and Classification Success Index values for Dataset 1 and Dataset 2 are given in Table 6. In our study, the performance metrics showcased successful outcomes.

TABLE VII
COMPARISON OF THE RECOMMENDED MODEL WITH EXISTING STUDIES

Study	Year	Accuracy (%)	F1-Score (%)
Mehnatkesh et al. [10] ³	2023	98.69	98.46
Deepak&Ameer [16] ³	2019	97.17	97.2
Turkoglu [24] ³	2021	98.04	97.95*
Alanazi et al. [28] ¹	2022	95.75	95.72*
Saurav et al. [26] ¹	2022	95.71	95.98
Kang et al.[29] ¹	2021	93.72	-
Ayadi et al. [18] ²	2021	98.49	98.3*
Aurna et al. [21] ²	2022	98.96	99
Gomez-Guzman et al. [23] ²	2023	97.12	97.28*
Proposed Model	Dataset-1 ¹	98.01	98
	Dataset-2 ²	99.62	99.62

¹Sartaj dataset <https://www.kaggle.com/sartajbhuvaji/brain-tumor-classification-mri>

²Masoud dataset <https://www.kaggle.com/datasets/masoudnickparvar/brain-tumor-mri-dataset?select=Training>

³Figshare dataset https://figshare.com/articles/dataset/brain_tumor_dataset/1512427

* F1-score is calculated from the confusion matrix or precision-recall values.

In Table 7, you can find a comparison between our model and prior studies. Some studies [10, 16 and 24] used the Figshare dataset in their study, which involved a three-class classification task (Glioma, Meningioma, and Pituitary). In contrast, other studies used datasets that incorporated normal

brain MRI images (four-class classification). When we compare our proposed model with other studies, it becomes evident that our model performs better. Transfer learning and ensemble learning methods have been widely used in existing studies. In addition, CNN models are used for feature extraction, and the resulting feature dataset is trained on a machine-learning algorithm for classification. Aurna et al. [21] used the two-stage method of feature extraction. They obtained an accuracy of 98.96% on Dataset 2. Apart from these, there are studies that propose modified CNN models. Ayadi et al. [18] proposed a customized CNN model with 10 convolutional layers. They tried to find the best optimizer and learning rate manually. Gomez-Guzman et al. [23] proposed a 17-layer CNN model with four convolution layers. However, they found the best result using the InceptionV3 model with transfer learning. Their accuracy on Dataset 2 is 97.12%. In Table 7, our proposed model also gives better results when compared with the studies in Dataset 1. In our proposed CNN model, we directly optimize the depth, width, and other hyperparameters in the CNN architecture using Gaussian process-based Bayesian optimization.

V. CONCLUSION

Brain tumors, a deadly type of cancer impacting both genders, have historically been diagnosed through risky biopsy procedures. Yet, the safer alternative of magnetic resonance imaging (MRI) has become more common in recent times. The main objective of this study is to differentiate brain tumor types from the MRI image, thereby guiding suitable treatment approaches. CNNs excel in autonomously extracting and classifying features from medical images, yielding favorable outcomes. Nevertheless, to ensure proficient learning within CNN architectures, a generous dataset is imperative. The focal issue revolves around pinpointing the most suitable depth and width hyperparameters that can facilitate optimal learning with the available limited data. Given the extended duration of the training phase and the considerable time investment required to explore all possible combinations, we adopted the Bayesian Optimization technique. This approach streamlined the process of identifying the optimal hyperparameter combinations. This study's focus was on determining the ideal hyperparameter configurations across two separate datasets. In Dataset 1, we reached an accuracy of 98.01% and an F1 score of 98%. In Dataset 2, our endeavors led to an impressive accuracy of 99.62%, also reflected in the F1 score. The model we have developed presents a valuable tool for clinicians, aiding in the precise identification of brain tumor types. As we look ahead, upcoming studies will explore diverse sets of hyperparameters and alternate datasets.

REFERENCES

- [1] Siegel, R. L., Miller, K. D., & Jemal, A. (2019). Cancer statistics, 2019. *CA: a cancer journal for clinicians*, 69(1), 7-34.
- [2] Bhatele, K. R., & Bhadauria, S. S. (2020). Brain structural disorders detection and classification approaches: a review. *Artificial Intelligence Review*, 53(5), 3349-3401.

- [3] Nazir, M., Shakil, S., & Khurshid, K. (2021). Role of deep learning in brain tumor detection and classification (2015 to 2020): A review. *Computerized Medical Imaging and Graphics*, 91, 101940.
- [4] Sharif, M. I., Li, J. P., Naz, J., & Rashid, I. (2020). A comprehensive review on multi-organs tumor detection based on machine learning. *Pattern Recognition Letters*, 131, 30-37.
- [5] Kaya, M. and Çetin-Kaya, Y. (2021). Seamless computation offloading for mobile applications using an online learning algorithm. *Computing*, vol. 103, no.5, pp. 771-799.
- [6] Miao, Y., Wu, G., Li, M., Ghoneim, A., Al-Rakhami, M., & Hossain, M. S. (2020). Intelligent task prediction and computation offloading based on mobile-edge cloud computing. *Future Generation Computer Systems*, 102, 925-931.
- [7] Rashed, A. E. E., Elmorsy, A. M., & Atwa, A. E. M. (2023). Comparative evaluation of automated machine learning techniques for breast cancer diagnosis. *Biomedical Signal Processing and Control*, 86, 105016.
- [8] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, 521(7553), 436-444.
- [9] Alzubaidi, L., Zhang, J., Humaidi, A.J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaria, J., Fadhel, M.A., Al-Amidie, M. and Farhan, L., (2021). Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *Journal of big Data*, 8(1), pp.1-74.
- [10] Mehnatkesh, H., Jalali, S. M. J., Khosravi, A., & Nahavandi, S. (2023). An intelligent driven deep residual learning framework for brain tumor classification using MRI images. *Expert Systems with Applications*, 213, 119087.
- [11] Liu, Z., Tong, L., Chen, L., Jiang, Z., Zhou, F., Zhang, Q., ... & Zhou, H. (2023). Deep learning based brain tumor segmentation: a survey. *Complex & intelligent systems*, 9(1), 1001-1026.
- [12] Krizhevsky, A., Sutskever I. and Hinton G. E. (2012). ImageNet classification with deep convolutional neural networks," *Proc - Neural Information Processing System Conference*, pp. 1-9.
- [13] Toğaçar, M., Ergen, B., & Cömert, Z. (2020). BrainMRNet: Brain tumor detection using magnetic resonance images with a novel convolutional neural network model. *Medical hypotheses*, 134, 109531.
- [14] Toğaçar, M., Cömert, Z., & Ergen, B. (2020). Classification of brain MRI using hyper column technique with convolutional neural network and feature selection method. *Expert Systems with Applications*, 149, 113274.
- [15] Balamurugan, T., & Gnanamanoharan, E. (2023). Brain tumor segmentation and classification using hybrid deep CNN with LuNetClassifier. *Neural Computing and Applications*, 35(6), 4739-4753.
- [16] Deepak, S., & Ameer, P. M. (2019). Brain tumor classification using deep CNN features via transfer learning. *Computers in biology and medicine*, 111, 103345.
- [17] Başaran, E. (2022). A new brain tumor diagnostic model: Selection of textural feature extraction algorithms and convolution neural network features with optimization algorithms. *Computers in Biology and Medicine*, 148, 105857.
- [18] Ayadi, W., Elhamzi, W., Charfi, I., & Atri, M. (2021). Deep CNN for brain tumor classification. *Neural processing letters*, 53, 671-700.
- [19] Ait Amou, M., Xia, K., Kamhi, S., & Mouhafid, M. (2022). A Novel MRI Diagnosis Method for Brain Tumor Classification Based on CNN and Bayesian Optimization. In *Healthcare* (Vol. 10, No. 3, p. 494). MDPI.
- [20] Alhassan, A. M., & Zainon, W. M. N. W. (2021). Brain tumor classification in magnetic resonance image using hard swish-based RELU activation function-convolutional neural network. *Neural Computing and Applications*, 33, 9075-9087.
- [21] Aurna, N. F., Yousuf, M. A., Taher, K. A., Azad, A. K. M., & Moni, M. A. (2022). A classification of MRI brain tumor based on two stage feature level ensemble of deep CNN models. *Computers in biology and medicine*, 146, 105539.
- [22] Kazemi, A., Shiri, M. E., & Sheikhhamedi, A. (2022). Classifying tumor brain images using parallel deep learning algorithms. *Computers in Biology and Medicine*, 148, 105775.
- [23] Gómez-Guzmán, M.A., Jiménez-Beristain, L., García-Guerrero, E.E., López-Bonilla, O.R., Tamayo-Pérez, U.J., Esqueda-Elizondo, J.J., Palomino-Vizcaino, K. & Inzunza-González, E. (2023). Classifying brain tumors on magnetic resonance imaging by using convolutional neural networks. *Electronics*, 12, 955.
- [24] Türkoğlu, M. (2021). Brain Tumor Detection using a combination of Bayesian optimization based SVM classifier and fine-tuned based deep features. *Avrupa Bilim ve Teknoloji Dergisi*, (27), 251-258.
- [25] Mondal, A., & Shrivastava, V. K. (2022). A novel Parametric Flatten-p Mish activation function based deep CNN model for brain tumor classification. *Computers in Biology and Medicine*, 150, 106183.
- [26] Saurav, S., Sharma, A., Saini, R., & Singh, S. (2023). An attention-guided convolutional neural network for automated classification of brain tumor from MRI. *Neural Computing and Applications*, 35(3), 2541-2560.
- [27] Turk, O., Ozhan, D., Acar, E., Akinci, T. C., & Yilmaz, M. (2022). Automatic detection of brain tumors with the aid of ensemble deep learning architectures and class activation map indicators by employing magnetic resonance images. *Zeitschrift für Medizinische Physik*. <https://doi.org/10.1016/j.zemedi.2022.11.010>
- [28] Alanazi, M.F.; Ali, M.U.; Hussain, S.J.; Zafar, A.; Mohatram, M.; Irfan, M.; Albarrak, A.M. (2022). Brain tumor/mass classification framework using magnetic-resonance-imaging-based isolated and developed transfer deep-learning model. *Sensors*, 22, 372
- [29] Kang, J., Ullah, Z., & Gwak, J. (2021). Mri-based brain tumor classification using ensemble of deep features and machine learning classifiers. *Sensors*, 21(6), 2222.
- [30] Nergiz, M. (2023). Classification of Precancerous Colorectal Lesions via ConvNeXt on Histopathological Images. *Balkan Journal of Electrical and Computer Engineering*, 11(2), 129-137.
- [31] Sartaj Bhuvaaji, Brain tumor classification (MRI). <https://www.kaggle.com/sartajbhuvaaji/brain-tumor-classification-mri>, 2020. (Accessed 1 Jan 2023).
- [32] Masoud Nickparvar, Brain Tumor MRI Dataset. <https://www.kaggle.com/datasets/masoudnickparvar/brain-tumor-mri-dataset?select=Training> (Accessed 5 Jan 2023).
- [33] Fernandes, V., Junior, G. B., de Paiva, A. C., Silva, A. C., & Gattass, M. (2021). Bayesian convolutional neural network estimation for pediatric pneumonia detection and diagnosis. *Computer Methods and Programs in Biomedicine*, 208, 106259.
- [34] Pelikan, M., Goldberg, D. E., & Cantú-Paz, E. (1999). BOA: The Bayesian optimization algorithm. In *Proceedings of the genetic and evolutionary computation conference GECCO-99* (Vol. 1, No. 1999).
- [35] Brochu, E., Cora, V. M., & De Freitas, N. (2010). A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*.
- [36] Bergstra, J., Bardet, R., Bengio, Y., & Kégl, B. (2011). Algorithms for hyper-parameter optimization. *Advances in neural information processing systems*, 24.
- [37] Frazier, P. I. (2018). A tutorial on Bayesian optimization. *arXiv preprint arXiv:1807.02811*.
- [38] Kaya, M., Ulutürk, S., Çetin Kaya, Y., Altıntaş, O., & Turan, B. (2023). Optimization of Several Deep CNN Models for Waste Classification. *Ahmet ZENGİN, Sakarya University, Türkiye, azengin@ sakarya. edu. tr*, 6(2), 91.
- [39] Koukoulas, S., & Blackburn, G. A. (2001). Introducing new indices for accuracy evaluation of classified images representing semi-natural woodland environments. *Photogrammetric Engineering and Remote Sensing*, 67(4), 499-510.

BIOGRAPHIES



MAHİR KAYA, He graduated from the Industrial Engineering Department of İstanbul Technical University in 2000. He received his M.S and Ph.D. degrees in 2010 and 2016, respectively from the Department of Information Systems at Middle East Technical University. His research field is machine learning, deep learning, mobile cloud computing,

and optimization. Currently, he works as an Assistant Professor in the Department of Computer Engineering at Tokat Gaziosmanpaşa University.