

Development and Implementation of an Advanced Brain Tumor Detection Algorithm Utilizing MATLAB for Enhanced Diagnostic Accuracy

RECEP AKPUNAR
Student, Department of EEE,
Abdullah Gül University,
recep.akpunar@agu.edu.tr
ID:110110256

EMIRHAN SAHINOGLU
Student, Department of EEE,
Abdullah Gül University,
emirhan.sahinoglu@agu.edu.tr
ID:2011011044

OGUZHAN ALASULU
Student, Department of EEE,
Abdullah Gül University,
oguzhan.alasulu@agu.edu.tr
ID:2111011051

Abstract— Brain tumor detection is a crucial domain in medical diagnostics because of the necessity for well-timed and correct identification of tumor types, which importantly affects patient treatment procedures and results. This project is about developing a strong brain tumor detection algorithm using MATLAB, through a comprehensive dataset and advanced image processing techniques. The dataset contains 3064 CT images collected from 233 patients, including three types of brain tumors: 1426 images of glioma, 708 images of meningioma, and 930 images of pituitary tumors. Firstly, preprocessing steps such as histogram equalization, erosion, opening, closing, and skull removal are implemented to improve image quality and prepare the data for further analysis. As a second step, the segmentation process is applied, containing suitable morphological operations to advance an operational masking algorithm for all brain images. In addition, this step comprises extensive feature extraction, where 35 features, including auto-correlation, sum variance, sum entropy, and entropy, are originated from the images. Finally, feature selection and classification are implemented using Waikato Environment for Knowledge Analysis (WEKA). Features are selected based on their interest, and the outputs are classified and validated using K-fold cross-validation. The results show that classifiers such as BayesNet, Bagging, and Logistic Regression accomplish good performance, showing the efficiency of the MATLAB algorithm in correctly detecting brain tumors. The importance of this project is in its potential to develop diagnostic accuracy and support doctors in making learned decisions, eventually enhancing patient care.

Index Terms— Brain tumor, MATLAB, Feature extraction, WEKA, Classification

I. INTRODUCTION

Brain tumors represent a important challenge in medical diagnostics due to their complex structure and the vital significance for accurately and well-timed detection. Brain tumors could be categorized into numerous types, with gliomas, meningiomas and, pituitary tumors as the most prevalent. Correct identification of these tumors is vital for adjusting suitable treatment plans, enhancing treatment durations, and reducing mortality rates. Developments in medical imaging technologies, such as “Computed Tomography (CT)” and “Magnetic Resonance Imaging (MRI)”, have provided to doctors the operational tools for

picturing brain structures. However, interpreting these images by hand can be time-consuming and tending to error due to the variability in tumor types and the complication of brain anatomy. Consequently, there is a rising requisite for software systems that could contribute to medical doctor by delivering consistent and correct tumor detection.

This project has an aim to develop an advanced brain tumor detection algorithm using MATLAB to improve correct identification through the arrangement of image preprocessing, segmentation, feature extraction, and machine learning methods. The study employs a dataset of 3064 images from 233 patients with 1426 images of glioma, 708 images of meningioma, and 930 images of pituitary tumors. During the preprocessing step, different image enhancement techniques, for instances, histogram equalization, erosion, opening, closing, and skull removal, are implemented to arrange the images for upcoming analysis. These steps are needed for decreasing noise and improving image contrast, hereby more accurate segmentation. The segmentation phase comprises applying morphological operations to build an effective masking algorithm that can precisely define tumor boundaries. The feature extraction process is conducted, where 35 different features following the segmentation, —including auto-correlation, sum variance, sum entropy, and entropy—are derived from the segmented images. These features deliver valuable data about the texture and intensity patterns within the tumors, which are important for accurate classification. The flowchart of the process steps is depicted in the Figure 1.

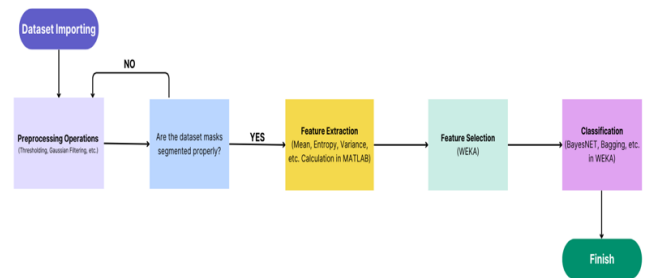


Fig.1 Illustration of a schematic diagram of brain tumor detection algorithm project

Subsequently, the feature selection and classification steps utilize the WEKA software to correlate the most appropriate features and implement machine learning algorithms to classify the tumors. Zainudin et al. (2019) explained that WEKA is the best-known open-source application in data-mining and machine learning algorithms. The performance of the classification models is validated using K-fold cross-validation, confirming the strength of the results. Classifiers such as BayesNet, Bagging, and Logistic Regression have shown talented performance with approximately %93 detection accuracy, showing the efficiency of the proposed methodology.

The significance of this project lies in its potential to improve the accuracy and efficiency of brain tumor detection, eventually supporting doctors in making informed decisions and improving patient care. Through advanced image processing and machine learning methodologies, this study contributes to the development of reliable and automated diagnosis tool in the domain of medical imaging.

II. ALGORITHM BUILDING AND PROCEDURES

This project presents a comprehensive and structured approach to algorithm development as well as procedural implementation. Initially, our designed model is aimed at creating an algorithm that more accurately and more reliably detects tumors on brain MRI images. This will lead to more accurate detection of patient outcomes. In the creation of this model, which is our project, feature extraction techniques that would be most suitable for our dataset of brain MRI images were applied and the most reliable techniques among the classification techniques were used. This results in the development of a sophisticated diagnostic tool that will greatly benefit medical professionals in the healthcare field by enabling more accurate and earlier detection of brain tumors. The methodical development process involves multiple steps, including data preparation, feature selection and classification. Moreover, all of these steps are completed in detail in order to increase the detection accuracy and optimize the performance of the model.

A. Preprocessing and Segmentation

The algorithm developed for our project uses preprocessing and segmentation methods to distinguish tumors from brain MRI images in the data used in our algorithm. In this way, it is aimed to achieve more reliable and accurate results in tumor detection. Initially, in order to process our data, the brain MRI images in our dataset were transferred to MATLAB application. These images are reliably stored in MATLAB's proprietary format and gray-scale processing is applied to the images by equalizing the pixel values so that the images are in a readable format. The images were then standardized by normalizing the pixel values to the range 0-255, making them suitable for further processing.

The next important steps are filtering and enhancement. A Gaussian blur filter was applied to each image to reduce the noise in the images and obtain a smoother image. This process made the tumors more prominent on the image. The intensity of

the blurring was controlled using the standard deviation feature. Another step involved applying greenish, bluish and purplish color tones to the grayscale images. This was achieved by blending different channels of the image to produce visually varied effects that are useful for identifying tumor regions on the image. In addition, contrast stretching was applied to the image using the “imadjust” function in the code written in MATLAB. This method mapped the intensity values of the input images to a specific range in the output image to increase the contrast of the grayscale images and make the features more distinct. In order to ensure accuracy in further processing steps, the enhanced images were converted to double precision and the pixel intensities of the images were normalized by compressing the pixel intensities of the images to the range [0, 1], thus facilitating more precise calculations and manipulations.

Segmentation involves the removal of skull and other non-brain tissues from brain images using a process known as skull stripping. This process starts by thresholding the images to separate brain tissue from other unnecessary structures as depicted in Figure 2. Then, a number of morphological operations are applied and binary images are cleaned.

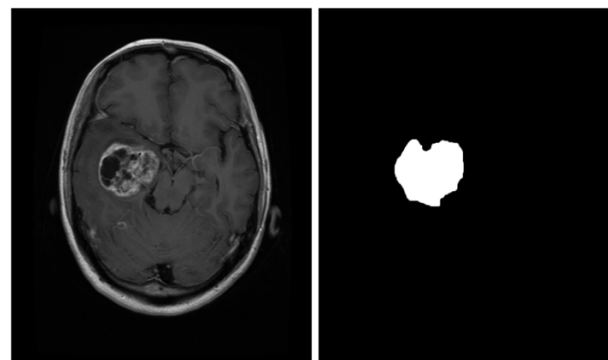


Fig.2 Original brain image and segmented tumor image after preprocessing and segmentation

Functions such as “bwareaopen” and “imfill” were used in coding in order to make the Brain MRI images in our dataset data used with the algorithm we designed smoother. In this step, small objects on the images were removed and gaps were filled. Immediately afterwards, a tumor mask is used to highlight and more clearly indicate the tumor regions within each of the brain MRI images. The “tumorMask” function was used to exclude non-tumor regions and leave only tumor areas of interest. The pixel intensities of the masked images used in our algorithm were then normalized. This process improved the appearance of the tumor regions on the images and made them clearly visible. It also simplified the process of analyzing the images. We also checked and evaluated the quality of our preprocessing steps by calculating the Peak Signal-to-Noise Ratio (PSNR), a metric that compares masked and processed grayscale images and evaluates the similarity between them. It also improves the image quality of the preprocessing and preserves important details in the images without affecting them. By correctly applying segmentation and preprocessing techniques on brain MRI images in our designed algorithm, the brain tumor images in our dataset will be suitable for use in the other steps of our

project and provide a solid foundation for reliable and accurate acquisition of brain tumors as shown in

B. Feature Extraction

Feature extraction, the next stage of our project algorithm for brain tumor detection, is crucial to identify and use the most reliable and informative features of brain MRI images to improve the accuracy of tumor detection. In this process, through our research, we identified the best performing and reliable feature extraction method for brain MRI images and implemented it for our algorithm. Thanks to this step, our analysis is successfully completed by making a more accurate classification on our images. When applying feature extraction techniques, many texture-based and statistical techniques are applied that highlight different aspects of the brain MRI images in our dataset and create a multifaceted view of the characteristics of the tumors in the brain MRI images.

Gray Level Coherent Matrix (GLCM) feature extraction technique, which is used to detect brain tumors more accurately and reliably, is known to be more effective than other feature extraction techniques. By examining the spatial relationship between the pixels of each brain image in our dataset, the GLCM technique obtains texture information that plays a major role in the process of distinguishing tumor tissues from normal tissues. As a result of the GLCM technique, basic features such as Entropy, Correlation, Energy, Homogeneity and Contrast are obtained. Among these basic features, Energy calculates the uniformity of the texture and is calculated by the process in equation (1):

$$Energy = \sum_{i=1}^N \sum_{j=1}^N P(i,j)^2 \quad (1)$$

The function $P(i,j)$ in this equation represents the normalized GLCM expression, while N is the gray level value. Entropy, another basic feature method, takes into account the image intensity distributions and calculates the randomness of these distributions and provides us with information about the complexity of the brain image texture, which can be computed as shown in equation (2):

$$Entropy = - \sum_{i=1}^N \sum_{j=1}^N P(i,j) \log (P(i,j)) \quad (2)$$

The higher the entropy value, the more complex textures are obtained. The contrast principal feature method also accounts for the local variations found in the GLCM technique. Furthermore, contrast is formulated as shown in equation (3) and identifies distinct intensity differences between neighboring pixels.

$$Contrast = \sum_{i=1}^N \sum_{j=1}^N j = 1^N (i-j)^2 P(i,j) \quad (3)$$

This basic feature method highlights the edges within the brain image. Correlation, another basic feature technique, calculates the linear dependence of the back levels in the co-occurrence matrix and is formulated as shown in equation (4):

$$Correlation = \sum_{i=1}^N \sum_{j=1}^N \frac{j = 1^N (i.j.P(i,j)) - \mu_x \mu_y}{\sigma_x \sigma_y} \quad (4)$$

In this equation, the means are defined as μ_x and μ_y and the standard deviations for the marginal distributions of $P(i,j)$ are expressed as σ_x and σ_y . The last of the basic feature extraction techniques, Homogeneity, analyzes the distance from the diagonal of the distribution of the components found in the GLCM method. Furthermore, Homogeneity expresses uniform textures as shown in equation (5):

$$Homogeneity = \sum_{i=1}^N \sum_{j=1}^N j = 1^N \frac{P(i,j)}{1 + |i-j|} \quad (5)$$

Since a high homogeneity value indicates a high similarity between pixels, it is understood that pixel values are like neighboring pixel values. Various features are calculated with the help of these mathematical operations to express the differences of the image texture. By carefully removing these features, we have greatly improved the capacity of our algorithm for brain tumor detection and have been able to improve our model to a reliable and advanced state for accurate and early detection. In our project, these operations for feature extraction have been used to properly distinguish tumor tissues from normal tissues in brain images. As a result, it plays a major role in achieving better patient outcomes.

C. Feature Selection

The feature selection step plays an important role in selecting the most appropriate features that provide a significant benefit to the efficiency and accuracy of this algorithm for detecting tumors in brain MRI images. The feature selection step reduces the size of the dataset by identifying the most informative features. As Alharan et al. (2019) indicated that the feature selection is a needed step to comprehend the performance of the upcoming materials such as classifiers. In addition, this process improves the performance of the model and minimizes the computational cost.



Fig.3 The segmented brain tumor image after merger of tumor mask and brain image

For our study, WEKA application was used to select feature extraction features. This application offers a wide range of features for statistical studies. As can be seen in Figure 3, correlation-based feature selection (CFS) was chosen as one of our most effective methods. The working logic of CFS is to

extract important features by using the redundancy value and individual predictive ability of each feature extraction feature. While doing this extraction, we make sure that the features are not collinear with each other.

The 10 features to be selected for classification include data on tumors as well as surrounding tissues, because the interaction of the brain tumor with surrounding tissues is as important as the structure of the tumor. The selected features are Inverse Difference Normalized (IDN), Homogeneity (Mean), Homogeneity (Prominence), Energy, Maximum Probability, Energy (GLCM), Information Measure of Correlation 1, Entropy (GLCM), Contrast (GLCM), and Correlation (GLCM). These feature extraction features were chosen because they are the most powerful methods for discriminating tumors by providing important information for accurate detection and classification of tumors.

<i>Extracted Feature Name</i>	<i>Correlation Rate</i>
Indirectness	0.1533
Homogeneity (Mean)	0.1605
Homogeneity (Prominence)	0.1591
Energy	0.4414
Maximum Probability	0.1829
Energy (GLCM)	0.1846
Information Measure of Correlation 1	0.0974
Entropy (GLCM)	0.1154
Energy from Gray Level Difference Matrix (GLDM)	0.1661
Homogeniety	0.1461

Table 1. Correlation Rates of Extracted Features

Among 35 extraction features, the effectiveness of the selected extraction feature features is given in the table (Table 1). Using the CFS method, the results of the features that are important for the dataset are plotted in a bar chart graph. With this graph, it is concluded which features are the most important. In this way, the model can be trained in a shorter time and more efficiently by avoiding the use of too many features for the classification process during model training because the selected features are the most important feature extraction features for the dataset. It should also be noted that the use of redundant features generally decreases the accuracy of the trained model and increases the training time of the model.

D. Classification

WEKA application was used in order to make classification. Five methods were selected for the classification application. Thornton et al. (2017) denoted that WEKA is operational at optimizing for certain objective functions but insufficient for allowing the user to determine that it fits model that simplify well. These are respectively; Logistic Regression (Logistic), Simple Logistic, BayesNet, Bagging and Classification Via Regression methods. Below are the explanation and detailed procedures of each title.

- **Logistic Regression (Logistic)**

Logistic regression method is a machine learning method that produces a value between 0 and 1 with a probabilistically non-

linear relationship between dependent variables and independent variables using the sigmoid function. For example, benign/malignant, sick/healthy, present/absent. In general, the threshold value is also used in this method and is 0.5 for this method. If the value produced is less than 0.5, it is placed in one class, if it is greater than 0.5, it is placed in the other class. Logistic regression method has 3 types.

- 1) **Binomial**

Results of this type can only be 0 or 1. There cannot be values in between. Therefore, there is no threshold value.

- 2) **Polynomial**

In this type, there are more than 2 dependent variable classes. For example, pen, paper and page. These classes are of unordered type.

- 3) **Ordered**

In this type of logistic, there are more than 2 classification categories, and they are ordered.

Concepts used in the Logistic Regression method

1) **Dependent Variable:** Variables that are tried to be predicted with the model created.

2) **Independent Variable:** Input variables for the prediction of the dependent variable. In the estimation process, help is received from independent variables.

3) **Logistic function:** It is the case of assigning a value between 0 and 1 for classification in the logistic regression model. A value between 0 and 1 is assigned to the dependent variables using independent variables.

4) **Log-odds:** Another name is logit function. It can be called the measure of the effect of independent variables on dependent variables. In other words, it indicates how much the change in the independent variable causes an increase or decrease in the dependent variable. In addition, if the log-odds value is less than 0, it causes a decrease in probability on the dependent variable, and if it is greater than 0, it causes an increase in probability.

5) **Coefficients:** Represents the value obtained in log-odds.

6) **Intercept:** indicates the basic probability of the categorized variable when the independent variables have no effect on the dependent variables.

7) **Maximum Likelihood Estimation - MLE:** It tries to calculate, in other words, to estimate the coefficient values to be used in the logistic regression method. Thus, the most appropriate parameters are selected for the model.

- **Simple Logistic**

Simple Logistic is an improved, slightly differentiated version of the Logistic method. Unlike the Logistic regression method, the regularization technique is used in the model. With the regularization technique, the model is further improved. Thus, it offers the possibility of working with a more accurate and wider dataset without increasing the complexity of the model and reducing the risk of overlearning. The same as Logistic Regression, the relationship between dependent variables and independent variables is established with the logit function and the model is expanded by reducing the risks by regularization. The other name of regularization is called L2 regularization. The coefficients of the model are squared and equalized to the penalty term. The penalty term is used in machine learning methods to reduce the complexity of the model by looking at the complexity of the model. Thus, the model is simplified. When the penalty term is large, it selects models with low

complexity, while if the penalty term is small, it selects complex models. Thus, it enables the model to be used with larger datasets. There are 3 types as L1 Penalty Term, L2 Penalty Term and Elastic Net.

- **L1 Penalty Term:** It is the summation of the coefficients of the model by taking their absolute values.

- **L2 Penalty Term:** It is obtained by summing the squares of the coefficients of the model.

- **Elastic Net:** It is formed by taking the combinations of L1 Penalty Term and L2 Penalty Term.

- **BayesNet**

BayesNet is a deep learning method that creates graphs based on probability. As the name suggests, it is a method based on Bayes theory. The model is developed and makes probabilistic inferences between the data. The model developed with training data is used to classify the data in the test set probabilistically. While making this classification, it uses causal relationships between the data. It provides great convenience in the classification process by specifying the probabilities graphically. While performing the classification process, it extracts the causal connections between them and makes inferences accordingly. A variable is expressed with a node on the graph. It shows the relationship between these nodes on the graph. Which variable affects which variables can be easily read on this graph. This process is called Directed Edge. Each node has a parent node and the child nodes have conditional connections to the parent nodes. This process is called Conditional Probability Distribution. It has several advantages thanks to its features. Since it is based on Bayes theorem, it is a model with a very high accuracy value. Since it establishes probabilistic reasons between the data, it performs the classification process even in the case of incorrect or missing data. In short, it works very well even in cases of uncertainty due to the complexity of the dataset.

- **Bagging (Bootstrap Aggregating)**

It is a machine learning method that is widely used in classification methods. It is also based on ensemble learning approach in machine learning. There is more than one type of bagging: Random Forest, Bagged Decision Tree, Bagged Neural Networks and Bagged Support Vector Machine. Weak learners algorithms are used for classification. Weak Learners methods can give very good results or even bad results when used alone, but these methods are used together and conclusions are drawn accordingly. Thus, even if there are bad algorithms, a model with high accuracy value is produced by using more than one method. First, random data are selected from the training dataset. These datasets are used to train the model and are called bootstraps. Secondly, weak learners models are trained using bootstraps and results are obtained. Finally, a result is obtained by combining each result using majority voting and averaging methods. According to this result, it is decided which class the data is in.

- **Classification via Regression**

In many classification methods, the classification process is performed in a discrete manner, but these processes may give incorrect results in the face of some data. In Classification via Regression, the classes are directly coded as numerical values such as 0 and 1 and these are taken as dependent variables. Regression model training is performed using dependent

variables. Training set is used as input data. With the trained model, the coded classes of the dependent variables are predicted. In order to do this, the data goes through several stages. First, as mentioned, the classifications are coded. These codes usually depend on the size of the classification but are commonly values such as -1 and 1. Secondly, the coded classes are separated as dependent variables. The data to be used in model training are independent variables. A model is trained using regression methods such as logistic regression. While training the model, the coefficients of the model are continuously optimized. Thus, after the model training is completed, the model is ready for the classification process. Then the data in the test set are entered into the model. A value between 0 and 1 is assigned for each data. If it is greater than the threshold value, it is classified into another class, if it is less than the threshold value, it is classified into another class. If the number of classes is greater than 2, the threshold value used during classification is different.

E. Evaluation

By using Logistic Regression (Logistic), Simple Logistic, BayesNet, Bagging and Classification Via Regression methods mentioned in the Classification section, the results were examined on the same dataset and tests and the most appropriate classification method was tried to be selected for our project. The following tables show the values of each classification method under different tests.

<i>Classification Method</i>	<i>Correctly Classified Instances</i>	<i>Time taken to build model</i>
BayesNet	95.8071%	0.08 seconds
Logistic	83.9623%	0.17 seconds
Simple Logistic	84.0671%	0.44 seconds
Bagging	93.6059%	0.24 seconds
Classification Via Regression	84.0671%	0.36 seconds

Table 2. Results obtained when the Percentage Split value is set to 66

<i>Classification Method</i>	<i>Correctly Classified Instances</i>	<i>Time taken to build model</i>
BayesNet	96.4349%	0.05 seconds
Logistic	81.9564%	0.13 seconds
Simple Logistic	81.9964%	0.24 seconds
Bagging	96.4349%	0.15 seconds
Classification Via Regression	83.066%	0.18 seconds

Table 3. Results obtained when the Percentage Split value is set to 80

<i>Classification Method</i>	<i>Correctly Classified Instances</i>	<i>Time taken to build model</i>
BayesNet	93.3713%	0.01 seconds
Logistic	84.0342%	0.11 seconds
Simple Logistic	84.248%	0.23 seconds
Bagging	85.3172%	0.13 seconds

Classification Via Regression	83.8917%	0.17 seconds
-------------------------------	----------	--------------

Table 4. Results obtained when the Percentage Split value is set to 50

Classification Method	Correctly Classified Instances	Time taken to build model
BayesNet	94.8326%	0.01 seconds
Logistic	83.5353%	0.1 seconds
Simple Logistic	83.464%	0.23 seconds
Bagging	93.2644%	0.13 seconds
Classification Via Regression	83.856%	0.17 seconds

Table 5. Results obtained when the cross-validation value is set to 10

Classification Method	Correctly Classified Instances	Time taken to build model
BayesNet	94.6187%	0.01 seconds
Logistic	83.5353%	0.1 seconds
Simple Logistic	83.464%	0.21 seconds
Bagging	93.9416%	0.13 seconds
Classification Via Regression	83.8917%	0.16 seconds

Table 6. Results obtained when the cross-validation value is set to 20

Classification Method	Correctly Classified Instances	Time taken to build model
BayesNet	94.9751%	0.01 seconds
Logistic	83.3215%	0.11 seconds
Simple Logistic	83.464%	0.22 seconds
Bagging	94.1554%	0.13 seconds
Classification Via Regression	83.9273%	0.17 seconds

Table 7. Results obtained when the cross-validation value is set to 50

F. Result and Discussion

When the results obtained are analyzed, BayesNet and Bagging methods have higher accuracy rates than the others among the 5 classification methods. In addition, optimization was performed in order to increase the accuracy rates obtained. With this optimization, it was aimed to find the optimum values of the required parameters so that the designed model could work in the best way. In this way, as can be seen in Table 8, the accuracy rates of our classification methods have increased and the system has achieved better performance. As the percentage split value increases, the model is trained with more data but has to be tested with less data. High accuracy predictions can be made with more trained models, but it should be noted that care should be taken about overlearning. For some models, an increase in accuracy is observed as the percentage split value decreases. It is concluded that an increase in the percentage split value for each model will not cause an increase in the accuracy rate. In addition, the use of more data in model training causes the system to spend more time in the model development

process. In the cross-validation process, for example, if the value is equal to 10, the number of data is divided into 10 classes to be equal. While 9 of the 10 classes are used to train the model, the remaining 1 class is used to test the model. When Table 5, Table 6 and Table 7 are analyzed, it cannot be said that the accuracy rate always increases as the cross-validation value increases. Since the model is trained with less data in each layer as the value increases with cross-validation, this may cause negative situations in some situations. For example, when the cross-validation value is 10, the models are more accurate than when the cross-validation value is 20, but when the cross-validation value is 50, the models can predict with more accuracy than the others. As a result, increasing the number of data for model training and decreasing the number of data for testing does not always lead to an increase in accuracy. In addition, increasing the number of layers in model training and decreasing the data used in model training in each layer does not always have a positive effect on the accuracy of the models.

The results obtained are very promising for future studies, but it should be noted that these systems can be further improved. Firstly, more number of feature extraction features can be extracted. Secondly, the developed model can be trained more by using a larger dataset. Thirdly, more classification methods can be used. By following such steps, the accuracy rate can be increased.

Test Options	Model Accuracy (%), Percentage Split (80% training)		Cross Validation
Classifier Methods	Before Optimization	After Optimization	Number of folds = 20
BayesNet	89.21%	96.43%	94.61%
Logistic	79.35%	81.95%	83.53%
Simple Logistic	71.32%	81.99%	83.46%
Bagging	82.93%	96.43%	93.94%
Classification Via Regression	70.17%	83.06%	83.89%

Table 8. Results obtained before and after optimization

III. Conclusion

In this study, by performing Segmentation, Feature Extraction, Feature Selection, Feature Selection, Classification and Performance Evaluation (Validation) steps, it has shown that it can predict brain tumors with the highest accuracy rate of 96.4349%. Compared to the traditional methods used in the past, the methods used in this study can both obtain faster results and predict with a higher accuracy rate. In the past, when the technology in the field of medicine was not developed, the detection of brain tumors and the classification of brain tumors took a lot of time and the accuracy of these steps was low, so such studies have been widely used in the field of medicine. The aim of this study is to shorten the time spent by doctors in determining the type of tumor and to provide an idea to the doctor in the decision-making process. In this way, a better quality service can be provided to patients by reducing the workload in the medical field but without a decrease in accuracy.

REFERENCES

- [1] Cheng, Jun, et al. "Enhanced Performance of Brain Tumor Classification via Tumor Region Augmentation and Partition." PloS one 10.10 (2015).
- [2] Cheng, Jun, et al. "Retrieval of Brain Tumors by Adaptive Spatial Pooling and Fisher Vector Representation." PloS one 11.6 (2016)
- [3] <https://www.geeksforgeeks.org/understanding-logistic-regression/>

- [4] Zainudin, Z., Shamsuddin, S. M., & Hasan, S. (2019). Deep learning for image processing in WEKA environment. *Int. J. Advance Soft Compu. Appl*, 11(1), 2074-282.
- [5] Alharan, A. F., Fatlawi, H. K., & Ali, N. S. (2019). A cluster-based feature selection method for image texture classification. *Indonesian Journal of Electrical Engineering and Computer Science*, 14(3), 1433-1442.
- [6] Thornton, C., Hutter, F., Hoos, H. H., & Leyton-Brown, K. (2013, August). Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 847-855).