

Enhancing Brain Tumour Detection based on Machine Learning Algorithms

Ahmed Hassan, Habiba Fathalla, Rahma Abdelkader, Youssr Hejy

ABSTRACT

MRI is the preferred imaging modality for brain tumour diagnosis due to its non-invasive nature and ability to provide accurate details about internal tumour information, but manual segmentation is time-consuming and prone to errors. Automated tumour detection methods are essential for accurate and timely diagnosis. Early detection is critical for better treatment outcomes, the motivation for this paper is to develop an automated brain tumour detection and segmentation system that can assist clinical experts in carrying out treatment in a more specific direction and improve the accuracy of diagnosis. We use three different brain magnetic resonance imaging (MRI) datasets that are public datasets, the proposed work is divided into three parts: preprocessing steps applied on brain MRI images, extracting texture features by using Shape-based features of the tumour and then classification model is done using machine learning algorithm as we used Random forest and XGboost with highest accuracy 91%.

1. INTRODUCTION

Brain tumors are a serious medical condition that can cause brain damage and be life-threatening. Early detection and classification of brain tumors are crucial for selecting the most appropriate treatment method to save patients' lives. Medical imaging, particularly MRI, plays a significant role in brain tumor diagnosis. However, manual tumor segmentation is a time-consuming, error-prone, and challenging task, which is not practical for analyzing large amounts of data. Therefore, the motivation for pursuing this problem is to develop automated tumor detection methods to improve diagnosis accuracy and save time for medical practitioners.

The human body is composed of various types of cells, each with a specific function. When some cells lose their ability to control their growth, they grow without any order, forming a mass of tissue called a tumor. Tumors can be benign or malignant, with malignant tumors leading to cancer. Magnetic Resonance Imaging (MRI) is a technique that plays a vital role in detecting brain tumors. It relies on the measurement of magnetic field vectors generated after appropriate excitation of strong magnetic fields and radio-frequency pulses in the nuclei of hydrogen atoms present in the water molecules of a patient's body. The MRI scan is much better than CT scan for diagnosis as it doesn't use any radiation, and radiologists can evaluate the brain using MRI. However, the conventional method for tumor detection in MRI images is human inspection, which is time-consuming and not appropriate for analyzing large amounts of data. Therefore, automated systems are needed to analyze the large volume of MRI data, which is more cost-effective and necessary for high accuracy when dealing with human life. Supervised techniques such as support vector machine, artificial neural networks, and unsupervised techniques such as fuzzy c-means and self-organization map (SOM) are commonly used to classify MR images as normal or abnormal.

Brain tumor Classification using machine learning involves training a model on a dataset of MRI images, along with corresponding **labels** indicating the presence or absence of a tumor. The trained model can then analyze new images and make predictions about the presence of a tumor. The model can also be used to classify different types of brain tumors, such as **gliomas**, **meningiomas**, and **pituitary** based on their **texture features**. This automated approach can significantly reduce the workload of radiologists and improve the accuracy of diagnosis, leading to better treatment outcomes for patients.

2. RELATED WORK

Vidyarthi. [1] This study proposes a machine learning-based approach for classifying the grade of malignant tumor types using a diverse feature set. The authors consider several high-grade malignant tumors, such as Central Neuro Cytoma (CNC), Glioblastoma Multiforme (GBM), Gliomas, Intra Ventricular Malignant Mass, and Metastasis, for classification in their premature stage of development. The proposed feature extraction mechanism uses both frequency and spatial domain features for classification of malignant tumors. Additionally, the authors identify relevant features using a cumulative frequency-based feature selection mechanism. To evaluate their proposed approach, the authors used (KNN) supervised classification algorithm for analysis of the results. For experimental purposes, they use 150 brain tumor MR images, with 30 images of each tumor type. The authors achieved a high classification accuracy of 94% for high-grade malignant tumors.

P.Balaji. [2] This study proposes a novel approach for the classification and detection of brain tumors using magnetic resonance images and a learning machine. The authors use a spatial fuzzy clustering algorithm for image segmentation, which allows them to detect brain tumors in their early stages and analyze anatomical structures. They also employ a probabilistic neural network with radial basis function for automated brain tumor classification which improve the accuracy and speed of diagnosis and use a Computer-Aided Diagnosis (CAD) system for early detection of brain tumors. The proposed method utilizes Dual Tree CWT multi-scale decomposition for texture analysis and Grey Level Co-occurrence Matrix (GLCM) for feature extraction.

But the paper lacks the discussion on the performance of their method on large datasets which affect the generalizability of their technique.

M. P. Aiwale and S. M. Ansari [4] The proposed method utilizes an OSTU algorithm for segmentation of tumor regions from an MR image. KNN and LLOYED algorithms are used for detecting and distinguishing tumor-affected tissues from non-affected tissues. The study also employs 12 features for extraction, such as correlation, contrast, energy, and homogeneity, which are obtained by performing wavelet transform on the converted gray scale image.

DB5 wavelet transform is used for feature extraction, it also has the ability to calculate the area occupied by the tumor in the overall brain cells. The study evaluates the performance of the proposed method using various metrics such as sensitivity, specificity, accuracy, and area under the curve (AUC).

3. Dataset and Features

This dataset is a combination of three different datasets [5], namely fig share, SARTAJ dataset, and Br35H. It contains a total of 7023 human brain MRI images that are classified into four different classes: glioma, meningioma, no tumour, and pituitary. The "no tumour" class images were taken from the Br35H dataset, while the remaining images were sourced from the figshare and SARTAJ datasets. This dataset can be used for various tasks such as brain tumour detection, segmentation, and classification.

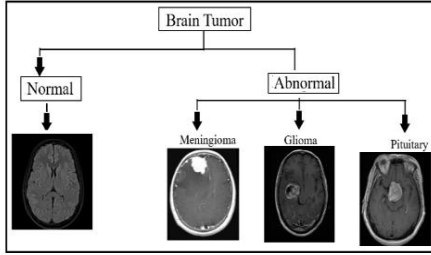


Fig 1. Sample of the Data

Our data files is divided into 2 big files Training and Testing files, From the training file we have 5,712 images, we split the data by applying: 20% validation data and 80% for training data, and the test file contains 1,311 images all kept for testing the model performance.

Preprocessing: Preprocessing is an essential step in image processing as it helps to improve the quality of the image, which in turn enhances the important features in the image and makes it suitable for further processing.

First we converted the image to grayscale to speed up the processing time, then we applied Gaussian Blur to the grayscale images to blur and reduce the noise in the images, then we binarize the images by applying threshold and also apply morphological operations (erosion and dilation) to remove any small regions or irregularities in the images and then on the resulted apply active counterimg and grab the largest contour and find the extreme points which correspond to the edges of the brain region and then crop the images

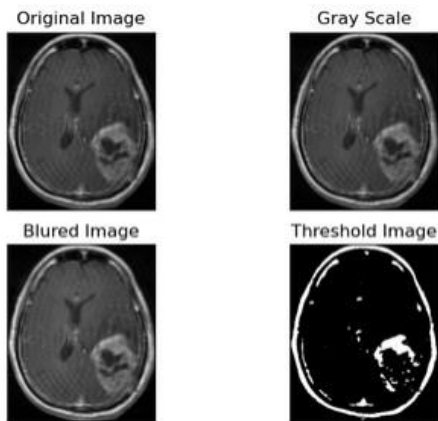


Fig 2. Preprocessed Images

Feature extraction is the process of transforming raw input data into a smaller set of representative features that can be used as input to a machine learning algorithm. This is necessary when the raw input data is too large or too complex to be processed directly, or when it contains redundant information or noise.

We extracted a lot of features which lately will be selected according to their importance for the model, First we extracted the **Mean** of the image which measures the average brightness of the image and also the **Standard deviation** which indicates how spread out the pixel values are from the mean by using root squared differences and the **variance** of the image which also indicates the spread out the pixel values are from the mean but by using squared differences and the **Skewness** which measures the asymmetry, "a positive skewness value indicates that the right tail of the histogram is longer than the left tail while negative skewness value indicates the opposite "

Then for calculating the **Entropy** which inform us of the uncertainty of the image we first need to compute the histogram of the image and the entropy is calculated using

$$H(X) = - \sum p(x) * \log_2(p(x)),$$

And then we calculated **Kurtosis** which measures the Preakness of image histogram, "a high kurtosis value indicates that the e histogram has a sharp peak, and the pixel values are concentrated around the mean."

And we extract **the HU Moments Invariants** which is a set of 7 numbers that are computed from the image moments which are used in image recognition and pattern matching but we only used the first 4 values cause they're related to the brightness, symmetry, and skewness of the image. The equations are as follows:

- Hu Moment 1:
 $(\mu_{20} + \mu_{02})$
- Hu Moment 2:
 $(\mu_{20} - \mu_{02}) + 4\mu_{11}$
- Hu Moment 3:
 $(\mu_{30} - 3\mu_{12})^2 + (3\mu_{21} - \mu_{03})^2$
- Hu Moment 4:
 $(\mu_{30} + \mu_{12})^2 + (\mu_{21} + \mu_{03})^2$

Where $(\mu_{20}, \mu_{02}, \mu_{30}, \mu_{03}, \mu_{12}, \mu_{21})$ are the normalized central moments of the image, which are derived from the raw image pixels.

Then we try different approach and we extract Texture Features from the segmented images using **Gray Level Co-occurrence Matrix (GLCM)** [6], which are used to show the texture property if an image and also used to distinguish between normal and abnormal brain images, which involves computing a matrix that describes the frequency of occurrence of pairs of pixel values at a given distance and angle in the image, we can extract them by using:

- Energy: which measure the textural uniformity

$$E = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} p(i, j)^2$$

- Contrast: measure of intensity contrast between a pixel and its neighbor

$$con = \sum_{i=0}^{n-1} n^2 * \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} p(i, j)^2$$

- Homogeneity: measure of closeness of distribution of elements in GLCM-to-GLCM diagonal

$$H = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} \frac{p(i,j)}{(1 + \text{mod}(i,j))}$$

- Correlation: a measure of how correlated a pixel to its neighbor

$$C = \frac{1}{\sigma_x \sigma_y} \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} (i,j) p(i,j)^2 - \mu_x \mu_y$$

Fig 3. Extracted Features Images

We also extracted features related to the tumor itself by extracting the contour of the tumor in the MRI images and the calculate what is called as Shape features which are the **Area** of the tumor and the **perimeter**, **circularity** which define how closely the contour resembles a circle $\text{circularity} = \frac{4\pi * \text{Area}}{(\text{perimeter})^2}$ and **convexity** which measure the area of the convex hull of the contour ,in another words it's the smallest polygon which encloses the contour and also calculated the **Solidity** which is defined as the ration between contour area and convex hull area ,also compute the **Major and Minor Axes** which are the lengths of the axes of the ellipse which best fit the contour of the tumor ,then we calculated **Eccentricity** which define how elongated the contour is and measured as the ratio of the distance between the foci and the major axis length which both define the best ellipse which fits the tumor contour.

These are all the extracted features which we thought may generalize or increase the performance of our models and then we applied features selection technique to find the most important features in them which will really effect our model.

1. PROPOSED METHODS

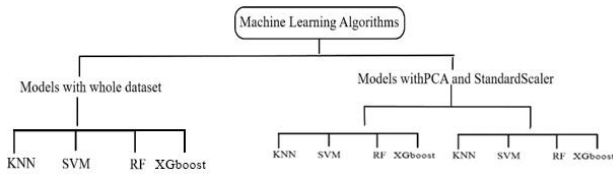


Fig 4. Proposed Method for Brain Tumor Detection in MR images

2. Model Methodology and loss functions

Random forest

It is an ensemble learning technique that combines multiple decision trees for making predictions. It aims to minimize the overall loss function, which is typically related to the accuracy or error of the predictions. The specific loss function used depends on the task, such as classification or regression. The algorithm constructs decision trees by recursively splitting the data based on selected features, aiming to reduce the loss function at each split. It utilizes bootstrap aggregating (bagging) to improve performance and reduce overfitting.

Bagging involves creating multiple bootstrap samples from the training data, training decision trees on each sample, and combining their predictions.

Loss functions: Gini impurity ($\text{Gini} = 1 - (p_1^2 + p_2^2)$), where p_1, p_2 are the probabilities of belonging to class 1, 2 respectively.

SVM with RBF Kernel:

It is a classification algorithm that aims to find the optimal decision boundary between classes. It uses the hinge loss function to measure classification error and optimize the model's performance. The RBF kernel computes the similarity between data points and determines their influence on the decision boundary. By tuning hyperparameters like C and gamma, SVM with RBF kernel can create a non-linear decision boundary and achieve high accuracy in classifying data.

Hinge Loss: $= \max(0, 1 - y * f(x))$

Hinge loss is used in SVM to measure the classification error. It penalizes misclassified samples based on the distance from the decision boundary. The goal of SVM is to minimize this hinge loss function while maximizing the margin between classes.

K-Nearest Neighbors (KNN):

It is a non-parametric classification algorithm that assigns a class label to a new data point based on the majority class among its k-nearest neighbors. It doesn't have an explicit loss function during training but uses a distance metric to determine the similarity between data points.

Distance Metric:

In KNN, the choice of distance metric plays a crucial role in determining the nearest neighbors. Common distance metrics include Euclidean distance, Manhattan distance, or Minkowski distance. The selected distance metric affects the model's performance and should be chosen based on the nature of the data and the problem at hand.

XGboost:

XGBoost (eXtreme Gradient Boosting) is an optimized implementation of the gradient boosting algorithm, which is efficient technique for classification.as it is known for its efficiency and high performance.

It's methodology involves creating an ensemble of weak prediction models, typically decision trees, and combining their predictions to make the final prediction. It follows a boosting approach, where each subsequent model is trained to correct the mistakes made by the previous models. This iterative process helps improve the overall predictive capability of the model.

Loss function: Softmax Loss (multinomial logistic regression)

which can be calculated as $\text{softmax}(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_j)}$ for j in all classes

where x_i represents the output score for class i , and $\exp(x_i)$ is the exponential of the score. The denominator is the sum of the exponentials of all class scores, ensuring that the probabilities sum up to 1.

4. Experiments and results

In this study, our objective was to evaluate and compare the performance of three distinct classification models: Support Vector Machine (SVM), k-Nearest Neighbor (KNN), and Random Forest (RF) for the purpose of brain tumor detection. An essential aspect of our modeling approach involved the meticulous tuning of hyperparameters, wherein we conducted a comprehensive grid search to identify the optimal parameter settings for each model.

For SVM, we determined that a penalty parameter value of "C=100" yielded the best results, along with a gamma value of 0.001, which is a critical parameter in the Radial Basis Function (RBF) kernel. In the case of KNN, we selected the number of neighbors to be "k=7" based on our analysis. As for Random Forest, we leveraged the default parameter values provided by the sklearn library, where the number of estimators was set to 100, and the criterion for splitting was defined as "gini," representing the Gini impurity criterion.

For XGboost we used the default values as max depth=6, n_estimator=100 and learning rate=0.1

Then after finding the best paramters for our models, we initially applied these models to our extracted features as a whole and obtained the following results:

Table 1. Models accuracies

Model	Train accuracy	Validation accuracy	Test accuracy
Random Forest	1.0	0.9	0.89
XGboost	1.0	0.92	0.91
RBF_SVM	0.52	0.53	0.49
KNN	0.67	0.6	0.57

From the initial results, we observed overfitting in both SVM and RF models, indicating that these models may be too complex for our dataset. To address this issue, we employed dimensionality reduction using Principal Component Analysis (PCA), however the accuracies of the test was extremely low so we applied feature scaling using StandardScaler to the dataset, The updated results are presented below:

Table 2. Models accuracies after scaling

Model	Train accuracy	Validation accuracy	Test accuracy
Random Forest	1.0	0.9	0.81
XGboost	1.0	0.92	0.86
RBF_SVM	0.88	0.86	0.82
KNN	0.86	0.84	0.8

After applying feature scaling, we observed a significant improvement in the performance of our models, indicating a reduction in overfitting and improved generalization capability.

Motivated by this progress, we decided to further enhance our models by performing feature selection. Given the large number of features in our dataset, it becomes crucial to

identify the most important features that significantly contribute to the modeling process. This feature selection step aims to improve model interpretability and reduce the complexity of the classification task.

In order to evaluate the impact of feature selection, we conducted experiments both before and after applying and standard scaling. The goal was to compare the differences in the results and assess the effectiveness of feature selection in enhancing the performance of our models.

First, we performed feature selection without applying scaling. The results are as follows:

Table 3. Models accuracies after features selection

Model	Train accuracy	Validation accuracy	Test accuracy
Random Forest	1.0	0.89	0.88
XGboost	1.0	0.9	0.9
RBF_SVM	0.62	0.61	0.58
KNN	0.75	0.69	0.66

Next, we incorporated scaling, and feature selection into our pipeline. The updated results are presented below:

Table 4. Models accuracies after features selection, and scaling

Model	Train accuracy	Validation accuracy	Test accuracy
Random Forest	1.0	0.89	0.77
XGboost	1.0	0.9	0.8
RBF_SVM	0.83	0.84	0.76
KNN	0.88	0.84	0.77

By combining scaling, and feature selection, we observed improvements in the performance of the XGboost, Random Forest and KNN models. However, the SVM model exhibited a slight decrease in test accuracy compared to the previous step. These results highlight the importance of finding the right balance between dimensionality reduction techniques and feature selection methods, as different models may respond differently to these transformations.

In conclusion, our experiments with feature selection demonstrated its potential in enhancing the performance of our classification models. By identifying the most relevant features, we achieved improved accuracy and interpretability. We will further analyze the selected features and explore their contribution to the modeling process.

Model Evaluation

Evaluating a classification model using accuracy, precision, recall, and F1-score provides a comprehensive understanding of its performance. These metrics help assess the model's ability to correctly classify instances from different classes and provide insights into its strengths and weaknesses.

Accuracy: Accuracy is the most commonly used metric and represents the ratio of correctly classified instances to the total number of instances. It is calculated as: $\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$, where TP (True Positives) are the correctly classified positive instances, TN (True Negatives) are the correctly classified negative instances, FP (False Positives) are the instances wrongly classified as positive, and FN (False Negatives) are the instances wrongly classified as negative.

Precision: Precision measures the proportion of correctly predicted positive instances out of all instances predicted as positive. It is calculated as: $\text{Precision} = \text{TP} / (\text{TP} + \text{FP})$

Precision is useful when the cost of false positives is high, and we want to minimize the number of false positives.

Recall: Recall, also known as sensitivity or true positive rate, measures the proportion of correctly predicted positive instances out of all actual positive instances. It is calculated as:

$\text{Recall} = \text{TP} / (\text{TP} + \text{FN})$, recall is important when the cost of false negatives is high, and we want to minimize the number of false negatives.

F1-score: The F1-score is the harmonic mean of precision and recall and provides a balanced measure of a model's performance. It is calculated as: $\text{F1-score} = 2 * (\text{Precision} * \text{Recall}) / (\text{Precision} + \text{Recall})$, the F1-score combines precision and recall into a single metric and is useful when we want to balance the trade-off between precision and recall.

The confusion matrix provides a visual representation of the model's performance by showing the counts of true positive, true negative, false positive, and false negative predictions for each class. It helps identify any patterns of misclassifications and provides insights into the model's behavior.

Results

These are the results we have obtained from evaluating our three models.

Table 5. Final evaluation

Model	Accuracy	Precision	Recall	F1-score
Random forest	0.77	1.0	1.0	1.0
XGboost	0.8	1.0	1.0	1.0
RBF_SVM	0.76	0.83	0.83	0.83
KNN	0.77	0.88	0.88	0.88

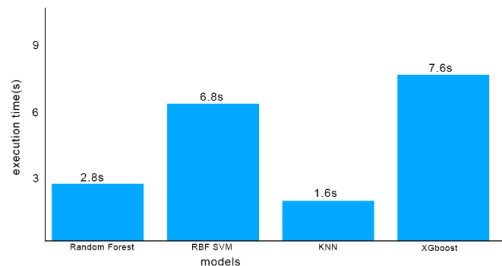


Fig 5. Models time execution

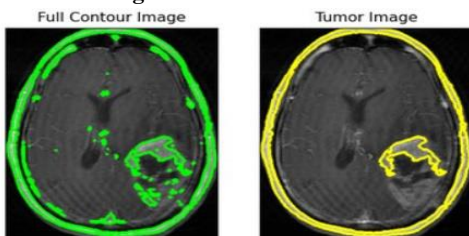
5. Conclusion

Through the analysis of the tumor feature extraction: We conclude that images need proper cropping on the brain only and the orientation won't cause any harm as long as we focus on tumor extraction.

We also found that no tumor class extract tumor features due to difference in image orientation, even after processing images, this causes our models to over-fit on tumor features and under-fit on no tumor class.

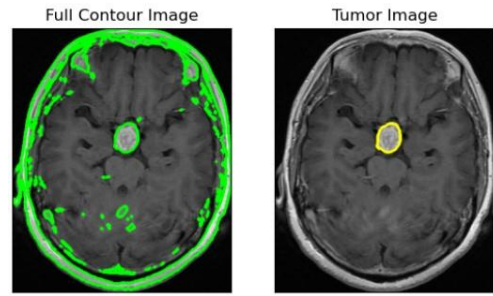
Glioma tumor shape is irregular and not clear in the images which cause some over detected tumor contours.

Fig 6. Glioma contour



Meningioma and pituitary are the most corrected extracted features from the tumor shape feature extraction.

Fig 7. pituitary contour



Since these struggles we found in feature extraction our future work can involve exploring the suitability and effectiveness of normalizing brain tumor images to the MNI 152 template. The MNI 152 template is a widely used standard template in neuroimaging research, representing a typical human brain. Normalizing brain images to this template involves aligning them spatially to a common anatomical space, allowing for better comparison and analysis across different subjects and studies.[7]

6. REFERENCES

6.1 Python Libraries

Numpy:

We used it in array creation: as provides functions to create multi dimensional arrays as well as array indexing and slicing as allows for advanced indexing and slicing operations on arrays.

Pandas

We used it in data representation: as it provides powerful data structures for efficient data handling and analysis as well as in data loading and writing as it offers functions to read data from various file formats, such as CSV,. It enabled us to load data into data frames, perform data transformations, and write data back to different file formats.

Matplotlib

We used it in plotting Data: pyplot provides a wide range of functions for creating different types of plots, such as line plots, scatter plots, bar plots, histogram plots, pie charts, and more. It allows you to visualize data and explore patterns or relationships between variables.

Seaborn

Statistical Data Visualization: Seaborn is specifically designed for statistical data visualization. It provides a high-level interface and a range of functions for creating attractive and informative statistical graphics so we used it in implementing confusion matrix integrated with Matplotlib.

Sklearn

Data Preprocessing: Scikit-learn provided us a variety of preprocessing techniques to prepare our data before training a model as it offers functions for data cleaning, feature scaling and handling missing values.

Machine Learning Algorithms: Scikit-learn provides a wide range of machine learning algorithms, including supervised and unsupervised learning algorithms. We used this library to implement SVM, RF and KNN.

Model Selection and Hyperparameter Tuning: Scikit-learn offers tools for model selection and hyperparameter tuning. It provided us with functions for cross-validation, grid search to find the optimal set of hyperparameters for your models. These techniques help in improving model performance and reducing overfitting.

Feature Selection: Scikit-learn provided us with methods for feature selection, allowing us to identify the most important features in your dataset.

6.2 Papers and articles

- [1] Vidyarthi, A., Agarwal, P., & Mittal, N.(2014, December 18)." Machine Learning based Classification of High Grade Malignant Brain Tumors using Diverse Feature Set".IEEE International Conference on Electronic Computer Technology
- [2] P. Balaji, Improving brain tumor classification on MRI using machine learning ..., Nov-2015.Brain Tumor Diagnosis Systems Based on Artificial Neural Networks and Segmentation Using MRI", IEEE International Conference on Informatics and Systems.
- [3] Jayachandran and K. Sunderaraj G, Brain tumor classification in MR images based on multi-modal texture.Jan-2015,IEEE International Conference on Informatics and Systems.
- [4] M. P. Aiwale and S. M. Ansari, (PDF) brain tumor detection using KNN - ResearchGate, Dec-2019. "MRI Brain Image Enhancement Using Filtering Techniques", International Journal of Computer Science & Engineering Technology.
- [5] D. Kavi, "Brain tumor image dataset," Kaggle, 07-Jul-2021.
- [6] GADKARI,D. (2000) Image Quality Analysis Using GLCM.
- [7] Nilesh Bhaskarrao Bahadure, Arun Kumar Ray, Har Pal Thethi, "Image Analysis for MRI Based Brain Tumor Detection and Feature Extraction Using Biologically Inspired BWT and SVM", *International Journal of Biomedical Imaging*, vol. 2017, Article ID 9749108, 12 pages, 2017.

7. Contribution

Member	Contribution
Ahmed Hassan	Modeling, hyperparametric tuning and feature extraction
Habiba Fathalla	Data processing, features selection and paper writing
Rahma Abdelkader	features selection , model evaluation and paper writing
Yours Ahsraf	Modeling, hyperparametric tuning and feature extraction