MRI Brain Tumor Classification Using HOG Features Selected *via* Impurity Based Importances Measure

Yasser Nizamli[1,2*] and Anton Filatov[2]

[1]*Tishreen University, Lattakia, Syria; yasser.nizamli@tishreen.edu.sy*
[2]*Saint Petersburg Electrotechnical University "LETI", Saint Petersburg, Russia; yanizamli@stud.etu.ru*

***Correspondence:** Yasser Nizamli; *yasser.nizamli@tishreen.edu.sy*

ABSTRACT- MRI is considered the primary method for confirming the diagnosis of brain tumors and choosing the appropriate treatment. Automating the process of detecting brain tumors in MRI images using deep models has become a popular trend in the scientific research community. However, deep neural networks require a large volume of data to avoid overfitting, which is not ideally available. This is where handcrafted features come in handy. In this paper, we present an efficient approach for brain tumor classification that can outperform deep CNN models. In the proposed system, the histogram of oriented gradients algorithm is used to extract feature descriptors from brain MRI images. The extracted features are processed using a random forest algorithm, where each decision tree performs the task of evaluating feature importances *via* the impurity metric, while all estimators collaborate in selecting the effective feature set. Finally, the Fine KNN algorithm is used to classify the types of brain tumors based on the numerical features obtained. The presented model achieved a high-test accuracy of 99.35% with an F1-score of 99.30%, outperforming many deep models.

Keywords: MRI; brain tumors; HOG; random forest; impurity; feature importances; fine KNN.

ARTICLE INFORMATION

Author(s): Yasser Nizamli and Anton Filatov;

Received: 10/08/2024; **Accepted**: 14/10/2024; **Published**: 30/11/2024;
e-ISSN: 2347-470X;
Paper Id: IJEER 1008-06;
Citation: 10.37391/ijeer.120416
Webpage-link:
https://ijeer.forexjournal.co.in/archive/volume-12/ijeer-120416.html

crossref member
CROSSREF.ORG
THE CITATION LINKING BACKBONE

Publisher's Note: FOREX Publication stays neutral with regard to Jurisdictional claims in Published maps and institutional affiliations.

1. INTRODUCTION

The brain acts as the centre of the nervous system responsible for controlling various functions of the human body. Any abnormal growth of cells within the tissue of this organ is likely to cause serious, life-threatening damage [1–3]. This abnormal growth is known as a brain tumour, and despite its many types, the majority of tumours, based on where they appear, can be classified into one of three basic categories: meningiomas, gliomas, and pituitary tumours [3–7]. Rapid detection is considered the barrier between life and death, while correct diagnosis helps in determining the appropriate treatment plan, which naturally leads to improving the patient's quality of life [2, 3, 8].

MRI is the most efficient and reliable method for diagnosing a brain tumour compared to other available techniques such as CT and PET [1, 3, 6]. MRI employs strong magnetic fields to generate a clear representation of brain tissue that is displayed by a computer. To complete the diagnosis process, the images must be examined by a trained expert who identifies the problem. This process is prone to error and consumes time and effort, in addition to the need for a lot of human resources in the first place [2, 5, 6]. To overcome these challenges, modern healthcare systems resort to automating diagnostic processes, taking advantage of the tremendous development in the field of artificial intelligence and machine learning [1, 3, 7, 9].

In automating the process of classifying brain MRI images using machine learning techniques, two main approaches can be observed: (1) based on hand-crafted features, (2) based on deep features. The first involves extracting features from MRI images using statistical methods that do not require parameter fitting. Among the most common of these methods are bag of features (BOF), Gray level co-occurrence matrix (GLCM), histograms of oriented gradients (HOG), and local binary pattern (LBP). The extracted features are then classified using classical machine learning algorithms such as SVM and decision tree [1, 2]. However, hand-crafted features involve a lot of redundancies, which increases the computational cost and reduces the effectiveness of classification. Therefore, it is necessary to find a suitable optimization technique to select the most effective features before passing them to a classifier [2].

The most widely used methods for extracting high-level features from brain MRI images are based on deep convolutional models. This option involves either building and training a deep neural network or using transfer learning approaches. Deep convolutional networks are developed from scratch or benchmark models can simply be retrained. In both cases the network consists of a series of convolutional blocks followed by a neural classifier [1, 4, 5, 7, 9–11]. Although this method does not use an additional feature optimization element, the huge number of parameters that need to be fitted requires a large amount of training data, which is difficult to provide in the healthcare field [3]. In contrast, the transfer learning approach relies on fixing the weights of the deep trained model and only fitting the top layer or replacing it with another classifier.

Among the most used models can be noted VGG, ResNet, DenseNet, and Inception [3, 6, 8, 12]. Like the case of hand-crafted features, deep extractors trained for different tasks and data sets produce a large number of features, which also requires an additional processing stage to prevent overfitting of the final classifier [3].

In this study we seek to overcome the issues addressed above, and therefore our major contributions can be summarized as follows:

To present a machine learning approach that achieves the advantages of high accuracy and simplicity in detecting abnormalities in brain MRI images.

To evaluate the effectiveness of HOG numerical features optimized using the impurity-based importances measure.

To evaluate the proposed approach using different metrics and provide comparison with models presented in the literature.

The rest of the paper is organized as follows: The Materials and methods section discusses the dataset used and its processing, as well as the components of the proposed system and how it works. The Results and discussion section presents the results of the experiment conducted with a comprehensive comparison with the state-of-the-art literature. Finally, the essence of the paper is summarized in a conclusion.

2. MATERIALS AND METHODS

In this study, the workflow of the proposed system can be summarized as follows:

- Brain MRI images are read from the dataset and pre-processed.
- A numerical representation of the MRI images is extracted using the HOG algorithm.
- A forest of decision trees is fitted to the numerical samples.
- The importance of each feature is evaluated based on the impurity values at the nodes of the trained trees.
- The set of features with high importances is passed to the Fine KNN classifier to perform the task of brain tumour classification.

2.1. Dataset and pre-processing

In this research to evaluate the proposed model, the well-known benchmark Fig share dataset containing 3064 brain T1-CE MRI images is used [13]. Data were collected from 223 patients and distributed into three categories representing common types of brain tumours: meningiomas, gliomas, or pituitary tumours. All images in the dataset have the same dimensions of 512 x 512 and are in grayscale format. *Figure 1* and *figure 2*. show examples of the dataset and its overall statistical distribution, respectively.

Minimal pre-processing is performed before passing the samples to the feature extraction stage. This involves normalizing the images and then resizing them to half the original dimensions in both width and height to reduce computational cost.

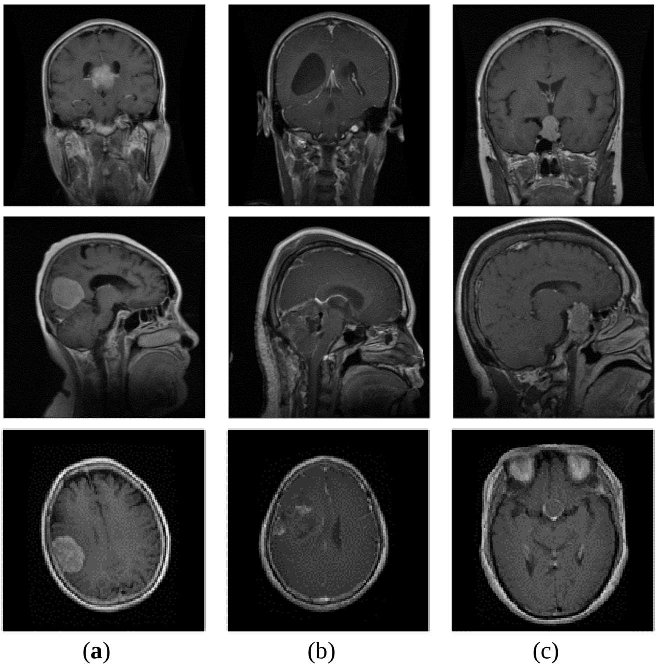


Figure 1. Examples from the dataset with different views: (a) meningiomas; (b) gliomas; (c) pituitary tumors.

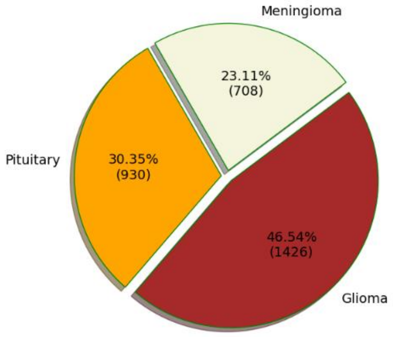


Figure 2. Data distribution

2.2. Feature extraction

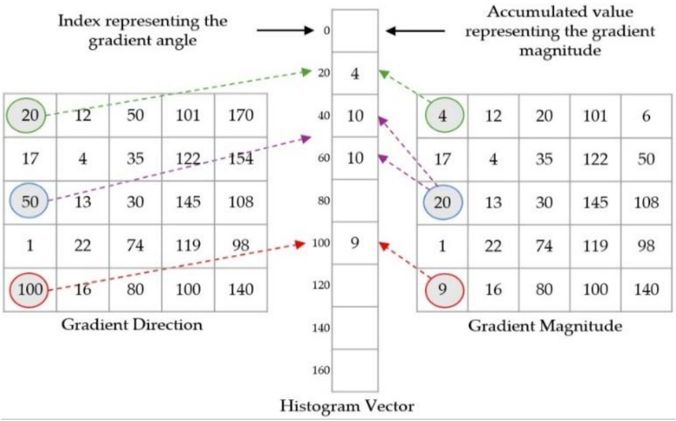
The histogram of oriented gradients method is used to perform the task of extracting numerical features from the brain MRI images. HOG is a statistical method that relies on calculating the distribution of horizontal and vertical gradients of an image [2]. This approach can describe any element or object in the image and thus produce features that are potentially useful for a classification task. Utilizing HOG can be summarized in the following steps [2]:

- The target image is divided into square cells of specified dimensions.
- The horizontal (g_x) and vertical (g_y) gradients for each cell are computed by applying derivative masks [-1, 0, 1] and [-1, 0, 1]^T, respectively. Using g_x and g_y the magnitude and direction of the gradient can be obtained as follows:

$$\text{gradient magnitude} = \sqrt{g_x^2 + g_y^2}$$

$$\text{gradient direction} = \arctan\frac{g_y}{g_x}$$

A histogram vector is generated for each image cell depending on the magnitude and direction matrices. The histogram bins are chosen based on the direction of the gradient, while the values within the bins are a cumulative weighted vote taken from the gradient magnitude. *Figure 3*. shows the process of filling the histogram bins.


Index representing the gradient angle
Accumulated value representing the gradient magnitude
Gradient Direction
Gradient Magnitude
Histogram Vector

20	12	50	101	170
17	4	35	122	154
50	13	30	145	108
1	22	74	119	98
100	16	80	100	140

0	
20	4
40	10
60	10
80	
100	9
120	
140	
160	

4	12	20	101	6
17	4	35	122	50
20	13	30	145	108
1	22	74	119	98
9	16	80	100	140

Figure 3. Histogram vector extraction

A specified number of cell histograms are grouped into a block whose values will be normalized. Various normalization methods can be used, including: L1-norm, L2-norm, L2-sqrt, and L2-Hys.

The process is repeated for all the blocks that slide to cover the entire image and obtain one large vector representing the extracted features.

Depending on the algorithm mechanism summarized above, four main parameters need to be adjusted, which are as follows: Orientations: Number of orientation bins. This parameter controls the length of the histogram vector formed for a cell. The value 9 is usually used as standard, which corresponds to the angle range from 0 to 180 with a step of 20 degrees.

Pixels per cell: Cell size in pixels. Cell size plays a very important factor in determining the size and quality of the resulting features. Choosing a cell size that is too small will provide a large number of features, which will lead to a jump in computational cost. Conversely, choosing a cell size that is too large will produce insufficient information that cannot accurately represent the shapes in the images. In this work, the cell size is chosen to be (5, 5) to ensure a balance between feature dimensions and effectiveness.

Cells per block: Number of cells in each block. To reduce illumination differences, it is preferable to normalize the histogram vectors. This process can be more effective if the normalization is performed for a set of cell histograms, usually called a block. However, using this method so that the block is in the form of a sliding window will greatly increase the number of features. The block size is chosen to represent only one cell (1, 1) thus preventing any intersections between histograms and limiting the size of the features.

Block norm: Block normalization method. In this research, L2**-**norm was chosen as it provided the best performance compared to other options.

The previous combination of parameter values adjustment ensures that features can represent the tumor region in MRI images while constraining the parameter size to some extent, which contributes to reducing complexity and limiting the effect of overfitting.

2.3. Feature optimization and classification

Features extracted by HOG are likely to be high-dimensional and contain a lot of redundancy, which in turn reduces performance and increases complexity [2]. To overcome this problem in our study, the feature importance measure expressed in the random forest algorithm was used for the purpose of selecting the most effective feature set. In this technique, an ensemble of decision trees is fitted to the training samples, where each estimator undertakes the task of evaluating the importance of the features separately based on the weighted impurity decrease computed at the nodes. The importance of a feature increases if its use to split a branch of a tree produces nodes that are purer, i.e. with a greater decrease in impurity, while the final importance of the feature is taken as an average of the values across all tree learners. A feature is ignored if its importance is less than a threshold chosen according to a specific criterion, otherwise it is kept [14, 15]. The following algorithm explains the feature importance evaluation mechanism using random forest.

# Algorithm 1

Define: L–Number of features; *M*–Number of training samples; *C*–Number of classes; *K*–Number of tree learners; im_{x_i} –Importance of feature x_i; $im_{x_{ik}}$ –Importance of feature x_i computed by tree t_k; *IM*–Importances vector; IM_{norm} –Normalized importances vector; n–Current selected tree node; (M_n; M_l; M_r)–Number of samples in node n, in the left child n_l, and in the right child n_r, respectively; (p_n^c; $p_{n_l}^c$; $p_{n_r}^c$)–Proportion of class c observations in node n, in node n_l, and in node n_r, respectively.

Input: K trained trees.
Output: $\mathrm{IM_{norm}}$.

Procedure:

```
// Start
For feature $x_i$, i = 0 to L-1 do
    For tree $t_k$, k = 0 to K-1 do
        For each node n in $t_k$ do
            If n involve $x_i$, then
                $\mathrm{impurity(n)} = \sum_{c=0}^{C-1} p_n^c(1 - p_n^c)$
                $\mathrm{impurity(n_l)} = \sum_{c=0}^{C-1} p_{n_l}^c(1 - p_{n_l}^c)$
```

```
7.  $\mathrm{impurity}(n_r) = \sum_{c=0}^{C-1} p_{n_r}^{c}(1 - p_{n_r}^{c})$
8.  $\mathrm{im}_{x_{ik}} \mathrel{+}= \frac{M_n}{M}\mathrm{impurity}(n) - \frac{M_l}{M}\mathrm{impurity}(n_l) - \frac{M_r}{M}\mathrm{impurity}(n_r)$
9.  End If
10. End For
11. End For
12. $\mathrm{im}_{x_i} = \frac{1}{K}\sum_{k=0}^{K-1} \mathrm{im}_{x_{ik}}$
13. Append $\mathrm{im}_{x_i}$ to IM
14. End For
15. $\mathrm{IM}_{\mathrm{norm}} = \frac{\mathrm{IM}}{\sum_{i=0}^{L-1} \mathrm{im}_{x_i}}$
// End
```

It is worth noting that in *step 8* of the algorithm the formula for calculating the weighted impurity decrease for a node is given, the accumulation of which across all nodes employing the specified feature will produce the importance value. Finally, the importances vector is normalized so that the sum of its values equals one, which may facilitate the process of choosing the appropriate threshold.

Applying feature importance measurement using random forest requires setting several parameters that can be presented as follows:

Number of estimators: The number of trees in the forest. The default setting for this parameter in skit-learn library is 100. Increasing the number of cooperating estimators may improve accuracy if the algorithm is used as a classifier. Although increasing the number of trees does not cause overfitting, it increases the computational cost significantly.

Max depth: The maximum depth of the tree. Increasing the depth of the tree to a large value (all leaf nodes are pure) will increase the computational cost and may lead to overfitting, while choosing a value that is too small may not achieve the required accuracy.

Bootstrap: Whether bootstrap sampling is used when building trees. If the value is False, the entire training data set will be used to build each tree, but it is important to know that this does not mean building identical trees as features are sampled when searching for the best split at each node.

The parameters were balanced to achieve the highest performance while avoiding increasing computational cost. A random forest of only twenty estimators was used with a maximum depth of ten layers. Training was performed without boosting while the threshold was chosen as the average of the feature importances.

To accomplish the task of classifying brain tumor types in numerically represented MRI images, the features that are considered important will be passed to the K-Nearest Neighbors (KNN) algorithm. KNN is one of the simplest and most widely used machine learning algorithms, as it works on a simple distance calculation process without any real parameter fitting. To classify a sample using KNN, the distance between this target sample and the rest of the data points is measured, and based on the votes of a specific number of neighbors K, the new sample is labeled. In our research, the decision is made based on the nearest neighbor of the target sample, i.e. the value of K is equal to 1, and the algorithm that adjusts in this way is called Fine KNN [10].

3. RESULTS AND DISCUSSION

To evaluate the developed model within the Google Colab CPU environment, 80% of the samples were allocated for training, while 20% for testing, which is a ratio commonly used in medical image studies [6]. *Table 1* shows a summary of the tuning parameters of the algorithms used to achieve the best performance.

Table 1. Parameters tuning.

Algorithm	Parameter	Setting
Histogram of oriented gradients	Pixels per cell	(5, 5)
	Number of histogram bins	9
	Cells per block	(1, 1)
	Block normalization	L2-norm
Impurity-based feature importances	Number of tree learners	20
	Maximum depth of a tree	10
	Bootstrap sampling	False
	Threshold	Mean of the feature importances
k-nearest neighbors	Number of neighbors	1
	Distance measure	Euclidean

Applying the importances measure that followed HOG led to reducing the number of features from 23409 to 3071, meaning that more than 85% of the features were considered unimportant, which will ensure a reduction in the computational cost. *Table 2* shows the classification report for the presented model while *figure 4* shows the confusion matrix (error matrix). A high classification accuracy of 99.35% was obtained, while the average F1-score reached 99.30%, demonstrating the absence of any performance bias towards one of the classes. An F1-score of 100% was achieved for the pituitary tumor class samples, while 100% recall (sensitivity) and 100% precision were obtained for the glioma and meningioma classes, respectively. The confusion matrix shows that only 4 samples were misclassified out of the entire test set, which is an error rate that does not exceed 0.65%.

Table 2. Classification report of the proposed approach

Category	Precision	Sensitivity	F1-score	Mean-F1	ACC
Meningioma	**100**	97.20	98.58	99.30	99.35
Glioma	98.65	100	99.32		
Pituitary	100	100	100		

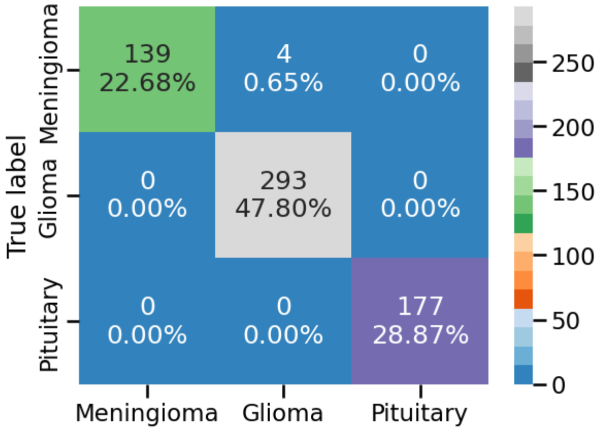


Figure 4. Confusion matrix of the proposed approach

The near-perfect performance of any machine learning model raises questions about overfitting and generalization issues. It is important to note that generalization failures are primarily caused by overfitting of the model, which occurs when high performance is achieved on the training dataset while poor performance is obtained on the test set. In our study, the results presented in *table 2* and *figure 4* are for the test dataset on which the model was not trained, indicating the effectiveness of the expert parameter tuning used to eliminate overfitting. Testing the model on an independent dataset can be a powerful addition to prove the reliability of the system, but due to the informality of other available datasets and their inconsistency with the three-class classification task, the study is limited to the Figshare dataset as is the case with published research. However, the adequacy of the dataset to ensure generalization can be assessed by plotting the model's learning curve for different percentages of dataset versus accuracy. In this approach, partial datasets are formed from the full dataset, starting with small subsets, and then increasing in size to include the entire original dataset. For each subset, the test accuracy and training accuracy are measured. If there is a gap between the two values, this indicates overfitting and thus the model is unable to generalize the results to new data, which is the case in which the need for a new larger dataset becomes critical to train the system.

Figure 5 shows the training curve of the proposed system. For a dataset size of less than 1000 samples, a large gap can be observed between the training accuracy value and the test accuracy value. As the dataset increases to more than 1500 samples, the test and training accuracy converge until they reach very close values when using the entire dataset. This discussion shows that the model presented in our research does not suffer from overfitting and meets the generalizability condition for the used dataset. It is worth mentioning that to create the learning curve, the accuracy measurement was based on 10-Fold Cross-Validation, and the model achieved an accuracy exceeding 99%, like the performance values calculated using the Train-Test-Split method, which is another important indicator of the system's robustness and performance strength, regardless of the part of the data that was tested on.

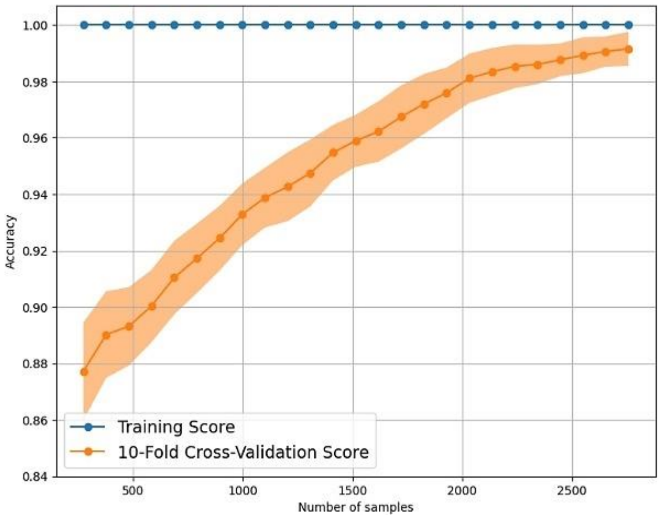


Figure 5. Learning curve of the presented model

To differentiate brain tumor types in MRI images, the Fine KNN algorithm was used due to its extreme simplicity. However, there are several other machine learning methods that may perform well as a final classifier, and thus it may be useful to validate the effectiveness of the model chosen in this research against potential competitors to confirm its superiority. The Fine KNN model is replaced by several well-known classifiers, namely linear Support Vector Machine (Linear SVC), Support Vector Machine with nonlinear kernel (SVC RBF), Logistic Regression (LR), Gaussian Naive Bayes (NB), and finally KNN with Three Neighbors (3NN).

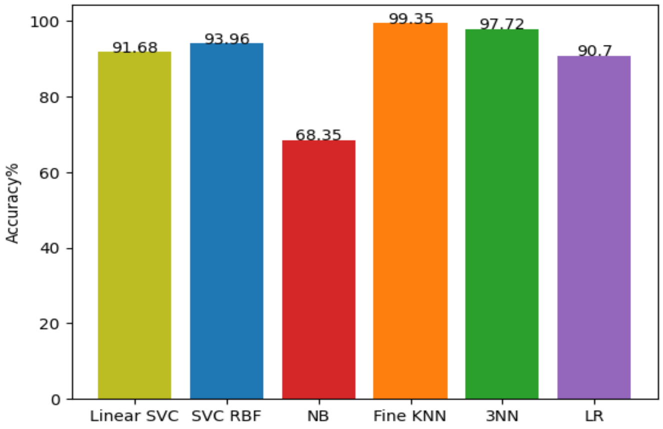


Figure 6. Performance of several machine learning algorithms as a final classifier

Figure 6 shows the performance of the machine learning classifiers compared to the used model. Naive Bayes fails to achieve acceptable performance, while the linear and nonlinear SVC models, in addition to Logistic Regression, have good classification performance of 91.68%, 93.96%, and 90.7%, respectively. The KNN models achieved the highest performance with a clear superiority of Fine KNN which was adopted as the final classifier in the overall system.

In addition to accuracy, the speed of inference and computational cost are essential performance metrics that determine the acceptability of the system for real-world use. Extracting features using a statistical method such as HOG ensures a reduction in computational requirements compared to deep models, while the assumed task of the feature selection stage is to process dimensions to reduce the time required to train and test and thus run the final classifier. *Table 3* shows the total time required by the classification algorithms in both the case of using feature selection and without it. It can be observed that using the feature processing step led to an effective reduction in both the training and testing time of all algorithms. The testing time or inference time is the most important factor because it represents the running state of the classifier, while the training process is usually done once for parameter optimization. As we mentioned in a previous section, the KNN algorithm does not require real parameter fitting since the inference process is straightforward. Using feature selection led to a reduction in the inference time using Fine KNN by more than five times, while the inference time can be reduced by about 15 times when using the SVC RBF algorithm compared to the case of not employing feature selection.

The added advantage of enhanced computational efficiency when using feature selection is very beneficial for any machine learning system especially when receiving large amounts of data in real-world operation.

Table 3. The effect of adding feature selection element on the execution time (sec) of classification algorithms

Algorithm	Approach			
	Without feature selection		With Feature Selection	
	Train	Inference	Train	Inference
Linear SVC	128.38	22.73	6.79	2.14
SVC RBF	158.38	58.44	14.27	3.91
NB	1.15	0.39	0.18	0.04
Fine KNN	–	2.65	–	0.47
3NN	–	5.48	–	0.63
LR	11.39	0.05	1.64	0.01

In *table 4* and *table 5* the proposed model is compared with those presented in scientific works in terms of performance and design and in terms of the pre-processing and augmentation techniques that were used, respectively. For the sake of valid comparison, works that use the same described dataset are discussed. Despite the simplicity of the design, the presented approach outperforms many models in terms of classification accuracy. In addition, considering that complex pre-processing operations [4, 8, 9] would increase the cost of the overall system due to the need to re-apply them upon deployment, the use of simple techniques has been adopted. Researchers in [1, 4, 6–10] employed various augmentation techniques for the purpose of increasing sample sizes and balancing classes, thus increasing accuracy, while in our study the results were obtained for the original dataset.

Table 4. Comparative analysis of literature in terms of design and performance

Ref., Year	Feature extraction	Feature optimization	Classification	ACC	F1-score
[1], 2022	CNN	–	SoftMax	93.10	–
	LBF	–	SVM	84.95	–
[3], 2024	VGG19	L1-feature selection + similarity fun.	SVM	98.53	98.34
[4], 2023	DCNN	–	SVM	96	96.92
[5], 2023	CNN	–	SoftMax	96	95.33
[6], 2024	DensNet169	–	Majority voting	95	95.10
[7], 2023	2 Parallel DCNNs	–	SoftMax	97.60	97.60
[8], 2023	VGG16	–	SoftMax	96.01	95.68
[9], 2023	YOLOv8s	–	SoftMax	–	92.47 [1]
[10], 2023	Deep feature fusion	–	SVM	95.4	94.93
	3 Parallel DCNNs	–	Majority voting	95.6	95.07
[11], 2022	CNN	–	SoftMax	97.2	96.5 [1]
Our model	HOG	Impurity-based feature importances	Fine KNN	99.35	99.30

[1] F1-score is not explicitly provided by the authors and is therefore calculated as a harmonic average of recall and precision.

Table 5. Comparative analysis of literature in terms of preprocessing and augmentation techniques

Ref., Year	Pre-processing	Augmentation
[1], 2022	Noise removal, resizing	Brightness adjustment, rotation, flipping
[3], 2024	Normalization, colorization, resizing, zero-centering	–
[4], 2023	Resizing, noise Reduction, contrast enhancement	Vertical and horizontal scaling, Horizontal and vertical shear, Horizontal and vertical translation
[5], 2023	Resizing, colorization	–
[6], 2024	Resizing	Rotation, image enlargement, flipping
[7], 2023	Resizing	Rotation, scaling, translation, filtering
[8], 2023	Neural autoregressive distribution estimation	Brightness adjustment, rotation, flipping
[9], 2023	Preprocessing is performed using the Roboflow platform (Not stated)	Rotation, horizontal flip, crop, shear, grayscale, brightness, exposure, noise
[10], 2023	Normalization, resizing	Generative adversarial network
[11], 2022	Normalization, resizing	–
Our model	Normalization, resizing	–

4. CONCLUSIONS

In this study, a machine learning approach is presented that aims to improve the efficiency of classifying tumor types in brain MRI images. After processing, a numerical representation is extracted from the MRI images by applying the histogram of oriented gradients technique. An ensemble of random forest decision trees then selects the most effective features using the importance measure based on the weighted impurity reduction calculations. The selected features are passed to the Fine KNN classifier to perform the task of detecting the type of brain tumor expressed in the MRI image. The proposed simple approach achieved high performance, outperforming many complex analogues presented in the literature, making it a promising option for further practical testing.

Supplementary Materials: The source code and results, along with extensive discussion, are provided *via* the following ***GitHub repository***: https://github.com/Yasser-A-Nizamli/Classification-of-tumors-in-brain-MRI-images.

Author Contributions: Conceptualization, methodology, validation, resources, writing—original draft preparation, writing—editing, Y.N.; supervision, A.F. All authors have read and agreed to the published version of the manuscript.

REFERENCES

[1] Sowrirajan, S. R.; Balasubramanian, S. Brain tumor classification using machine learning and deep learning algorithms. IJEER 2022, Volume 10, pp. 999–1004.

[2] Alavi, S. E.; Zare, E.; Rashti, M. j. Brain Tumors Detection on MRI Images through Extracting HOG Features. JOAASR 2018, Volume 2, pp. 9–25.

[3] Nizamli, Y.; Filatov, A. Improving transfer learning performance for abnormality detection in brain MRI images using feature optimization techniques. 2024 XXVII International Conference on Soft Computing and Measurements (SCM), Saint Petersburg, Russia, 2024, pp. 432–435.

[4] Biswas, A.; Islam, M. S. A hybrid deep CNN-SVM approach for brain tumor classification. JISEBI 2023, Volume 9, pp. 1–15.

[5] Jaspin, K.; Selvan, S. Multiclass convolutional neural network-based classification for the diagnosis of brain MRI images. Biomedical Signal Processing and Control 2023, Volume 83.

[6] Khan, S. U. R.; Zhao, M.; Asif, S.; Chen, X. Hybrid-NET: A fusion of DenseNet169 and advanced machine learning classifiers for enhanced brain tumor diagnosis. IMA 2024, Volume 34.

[7] Rahman, T.; Islam, M. S. MRI brain tumor detection and classification using parallel deep convolutional neural networks. Measurement: Sensors 2023, Volume 26.

[8] Sowrirajan, S. R.; Balasubramanian, S.; Raj, R. S. P. MRI brain tumor classification using a hybrid VGG16-NADE model. Brazilian Archives of Biology and Technology 2023, Volume 66.

[9] Passa, R. S.; Nurmaini, S.; Rini, D. P. YOLOv8 based on data augmentation for MRI brain tumor detection. Scientific Journal of Informatics 2023. Volume 10.

[10] Deepak, S.; Ameer, P. M. Brain tumor categorization from imbalanced MRI dataset using weighted loss and deep feature fusion. Neurocomputing 2023, Volume 520, pp. 94–102.

[11] Kibriya, H.; Masood, M.; Nawaz, M.; Nazir, T. Multiclass classification of brain tumors using a novel CNN architecture. Multimedia Tools and Applications 2022, Volume 81, pp. 29847–29863.

[12] Nizamli, Y.; Filatov, A.; Fadel, W.; Shichkina, Yu. Accurate Anomaly Detection in Medical Images using Transfer Learning and Data Optimization: MRI and CT as Case Studies. 2024 V International Conference on Neural Networks and Neurotechnologies (NeuroNT), Saint Petersburg, Russia, 2024, pp. 170–173.

[13] Brain tumor dataset. Available online: https://figshare.com/articles/dataset/brain_tumor_dataset/1512427/5 (accessed 26.03.2024).

[14] Zhang, B.; Dong, X.; Hu, Y.; Jiang, X.; Li, G. Classification and prediction of spinal disease based on the SMOTE-RFE-XGBoost model. PeerJ Computer Science 2023, Volume 9.

[15] Ahn H. Performance evaluation of a feature-importance-based feature selection method for time series prediction. JICCE 2023, Volume 21, pp. 82–89.