Biomedical Signal Processing and Control 104 (2025) 107538

Contents lists available at ScienceDirect

Biomedical Signal Processing and Control

journal homepage: www.elsevier.com/locate/bspc

FlexiCombFE: A flexible, combination-based feature engineering framework for brain tumor detection

Ilknur Tuncer [a], Abdul Hafeez Baig [b], Prabal Datta Barua [c], Rena Hajiyeva [d], Salvi Massimo [e,*], Sengul Dogan [f], Turker Tuncer [f], U.R. Acharya [g]

[a] Elazig Governorship, Interior Ministry, Elazig, Turkey
[b] School of Business, University of Southern Queensland, West Street Toowoomba, QLD, Australia
[c] School of Business (Information System), University of Southern Queensland, Australia
[d] Western Caspian University, Department of Information Technologies, Baku, Azerbaijan
[e] Biolab, PoliToBIOMed Lab, Department of Electronics and Telecommunications, Politecnico di Torino, Corso Duca Degli Abruzzi 24, Turin 10129, Italy
[f] Department of Digital Forensics Engineering, Technology Faculty, Firat University, Elazig, Turkey
[g] School of Mathematics, Physics and Computing, University of Southern Queensland, Springfield, Australia

ARTICLE INFO

Keywords:
Self-organized feature engineering
Feature extraction
Feature selection
Classification
Biomedical image classification

ABSTRACT

Deep learning models are prevalently employed for image classification, but significant opportunities remain within the domain of feature engineering. This study introduces FlexiCombFE, a novel, flexible, combination-based feature engineering framework for brain tumor detection using various fixed-size patch divisions. This framework employs three distinct feature extractors (Local Phase Quantization, Local Binary Pattern, and Pyramidal Histogram of Oriented Gradients) to generate a total of seven primary feature vectors. By using four types of fixed-size patch divisions, 28 feature vectors are generated. Then, three feature selectors (Chi-squared, Neighborhood Component Analysis, and ReliefF) create 84 selected feature vectors. In the classification phase K-nearest neighbors and support vector machine classifiers yield 168 classifier-specific outcomes. An information fusion generates 166-voted outcomes, with the most accurate classification outcome selected as the final output. This self-organizing feature engineering model achieved a classification accuracy of 99.35% on a brain tumor image dataset, outperforming several deep learning approaches. The modular design of the proposed framework allows for detailed analysis of the classification effects of individual methods, providing optimal feature engineering strategies for medical image classification tasks.

1. Introduction

Brain tumors occur when unusual cells grow in the brain, especially due to age [1]. As these tumors grow in the cells, the brain's pressure may increase, which can negatively affect human health and even cause permanent damage [2]. Brain tumors can be benign or malignant. Benign tumors usually do not spread and grow slowly with clear edges. Malignant tumors spread and grow rapidly and can damage brain functions [3]. Factors such as genes, environment, age, gender, immune problems, chemicals, and lifestyle can cause brain tumors [4,5] . However, brain tumors can also occur in people who do not have these risk factors [6,7]. According to World Health Organization data, 246 thousand people die from brain tumors every year in the world. At the same time, according to WHO, when mortality rates are examined, it ranks 12th among all cancer types [8]. Brain tumors can be seen in different types depending on the starting point and cell type. Primary brain tumors arise from brain tissue. Common primary brain tumors are meningiomas, pituitary adenomas and gliomas [9,10].

Just like with all types of cancer, early diagnosis of brain tumors is crucial in diagnosing and treating the disease [11]. Brain tumors are complex illnesses that are difficult to diagnose and treat. Diagnosing brain tumors requires a multidisciplinary approach [12]. Additionally, parallel to the advancement of technology, the data obtained by medical imaging systems play a vital role in diagnosing the disease [13,14].

* Corresponding author.
E-mail addresses: ilknur.tuncer@icisleri.gov.tr (I. Tuncer), Abdul.Hafeez-Baig@usq.edu.au (A.H. Baig), Prabal.Barua@usq.edu.au (P.D. Barua), rena.haciyeva@wcu.edu.az (R. Hajiyeva), massimo.salvi@polito.it (S. Massimo), sdogan@firat.edu.tr (S. Dogan), turkertuncer@firat.edu.tr (T. Tuncer), Rajendra.Acharya@usq.edu.au (U.R. Acharya).

https://doi.org/10.1016/j.bspc.2025.107538
Received 31 October 2024; Received in revised form 30 December 2024; Accepted 19 January 2025
Available online 26 January 2025

However, the abundance of data obtained and the necessity for rapid analysis indicate the need for reliable computer-aided diagnosis systems [15]. Manually examining these images by expert doctors is time-consuming and complex [16]. Every day, new machine-learning approaches are being suggested for the automatic, quick, and error-free evaluation of these diseases [17].

Despite the prevalence of deep learning models in image classification, there remains a significant gap in the development of advanced feature engineering techniques. Most existing approaches rely on fixed descriptors and patterns, lacking the flexibility to adapt to varying image characteristics. Furthermore, there is a scarcity of feature engineering models that can compete with the performance of deep learning approaches while maintaining interpretability. These limitations highlight the need for a more versatile and powerful feature engineering framework.

In this study, the FlexiCombFE model introduced a novel feature engineering framework specifically tailored for brain tumor detection. By employing a unique combination of patch-based feature extraction methods, including Pyramidal Histogram of Oriented Gradients (PHOG) [18,19], Local Phase Quantization (LPQ) [20] and Local Binary Pattern (LBP) [21], this research proposed a flexible and comprehensive approach to handling image data. The model's architecture allowed for generating 28 distinct feature vectors through various fixed-size patch divisions, further refined by three feature selectors (NCA [22], Chi2 [23], and RF [24] to produce 84 selected feature vectors. Applying k-nearest neighbors (kNN) [25] and support vector machine (SVM) [26] classifiers on these vectors led to an expansive set of 168 classifier-specific outcomes, ultimately synthesized through an information fusion process to select the most accurate classification result.

This framework significantly filled a gap in the literature by offering a deep-learning-competitive, yet inherently flexible, approach to feature engineering in the context of brain tumor image classification. This framework demonstrates the untapped potential of combining traditional feature engineering methods with modern classification techniques. The FlexiCombFE model demonstrated a high classification accuracy of 99.35 % on a public brain tumor image dataset, providing a compelling alternative to deep learning models, especially in scenarios where interpretability is crucial.

1.1. Related works

Table 1 summarizes recent studies on the classification of brain tumors, showing the diversity of approaches in this field. These studies typically aim to distinguish between two to four tumor classes, employing various methodologies from traditional machine learning to advanced deep learning techniques. While some researchers utilize classic approaches like gradient boosting and Support Vector Machines (SVM), the field is increasingly dominated by deep learning methods, particularly Convolutional Neural Networks (CNN) [27,28]. Performance is typically evaluated using metrics such as accuracy, precision, recall, and F1 score. Most methodologies achieve high accuracy rates, often exceeding 95 %.

Data augmentation techniques are frequently employed to enhance model performance, and various data-splitting strategies are used across studies. The choice of dataset and the number of classes varies, affecting result comparability. Recent approaches also focus on model interpretability, addressing a critical need in medical applications [29].

1.2. Literature gaps

- Most recent research has focused on deep learning models for image classification, leaving a scarcity of innovative feature engineering approaches in the literature.
- Existing feature engineering models generally use fixed descriptors and patterns. While deep learning models have employed multilevel or patch-based feature extraction methodologies, there are limited feature engineering models that incorporate such flexible approaches.
- There is a notable lack of feature engineering models that can match or exceed the performance of deep learning models while maintaining interpretability and efficiency.

1.3. Motivation and study outline

Accurate classification of brain tumors is crucial for precise diagnosis and treatment planning. Moreover, accurate classification is vital for clinical decision-making processes. A reliable classification model reduces diagnostic errors and increases confidence in medical decisions. The results obtained support radiologists and clinicians by providing early detection, which is essential for improving survival rates and reducing complications. Intelligent systems contribute to personalized treatment strategies by enhancing diagnostic accuracy. These systems help patients receive targeted treatments based on specific tumor characteristics. Therefore, in this research, we aimed to propose a lightweight and accurate automated brain tumor classification framework.

Variable deep learning models in the literature have been used in different methods to achieve high classification performances [30,31,32]. Since 2017, transformers have become a very popular research area in deep learning, particularly for natural language processing (NPL) applications [33]. Vision transformers (ViT), proposed by Dosovitskiy [34], demonstrated that transformers can be effectively utilized for image classification. ViT achieved higher classification performance compared to convolutional neural networks (CNN) [35]. After that, transformers have dominated the field of computer vision [36,37]. Initially, ViT has used fixed-size patches of 14×14, 16×16, or 32×32, with patch sizes significantly affecting the classification performances of the model [38]. To solve this problem, researchers have proposed FlexiViT models, which employ multiple patch sizes to solve image classification problems more effectively.

In this work, we have been inspired by the FlexiViT.We have used four fixed-size patches to generate different local features. To extract features, we used three feature extractors to get different feature vectors: (i) LBP, (ii) LPQ, and (iii) PHOG. Using the combinations of these feature extractors, we have extracted 7 ($=2^3$-1) feature vectors for each patch size. Thus, the proposed FlexiCombFE extracted a total of 28 ($=7 \times 4$) feature vectors.

In the feature selection phase, we used three feature selection functions since each feature selection function had its individual feature selection ability. Herein, we employed three commonly used feature selectors: (1) NCA, (2) Chi2, and (3) ReliefF. By applying these feature selectors, we have created 84 ($=28 \times 3$) selected feature vectors. For classification results, we used two classifiers: (1) kNN and (2) SVM. Our proposed FlexiCombFE generates 168 ($=84 \times 2$) classifier-wise outcomes in the classification phase. Then, we have analyzed the methods' classification performances using these classifier-wise outcome results. The final phase of the presented FlexiCombFE is information fusion. In this phase, we used iterative majority voting (IMV) [39] to obtain the voting results. Moreover, we selected the best outcome by employing a greedy algorithm.

Based on the above aspects, our main motivations are:

- Proposing a flexi-based feature engineering model;
- Demonstrating the high classification effect of the flexi and combination-based feature extraction;
- Obtaining a feature engineering framework;
- Testing the presented FlexiCombFE using a public image dataset.

1.4. Novelties and contributions

The proposed FlexiCombFE framework introduces several key innovations in feature engineering for medical image analysis, with the

Table 1
Related works.

Study	Method	Classifier	Dataset / Number of classes	Split ratio	Augmentation	The results (%)
Ghosh and Kole [41]	Feature extraction	Gradient Boosting, XGBoost	Dataset A 1465 images. 2 classes (brain tumor, no-tumor) Dataset B 2870 images. 4 classes (glioma, meningioma, pituitary, no tumor)	75:25	No	Dataset A Acc: 92.40 Pre: 85.00 Rec: 94.40 F1: 89.50 Dataset B Acc: 90.00 Pre: 90.00 Rec: 90.00 F1: 90.00
Rahman and Islam [27]	Parallel deep CNN	Softmax	Dataset A 253 images.2 classes (brain tumor, no-tumor) Dataset B 3064 images. 3 classes (glioma, meningioma, pituitary) Dataset C 2870 images. 4 classes (glioma, meningioma, pituitary, no tumor)	90:10	Yes	Dataset A Acc: 97.33 Dataset B Acc: 97.60 Dataset C Acc: 98.12
Patil and Kirange [28]	Ensemble deep CNN	Softmax	3064 images. 3 classes (glioma, meningioma, pituitary)	10-fold CV	Yes	Acc: 97.77 Pre: 96.66 Rec: 98.30 Sen: 96.66 Spe: 98.33 F1: 97.47
Saeedi et al. [42]	CNN	Softmax	3264 images. 3 classes (glioma, meningioma, pituitary)	90:10	Yes	Acc: 96.47 Pre: 94.75 Rec: 95.75 F1: 95.00
Aamir et al. [43]	Deep feature extraction	SVM	3064 images. 3 classes (glioma, meningioma, pituitary)	5-fold CV	Yes	Acc: 98.98
Hossain et al. [44]	IVX16	MLP	3264 images. 3 classes (glioma, meningioma umor, pituitary)	80:10:10	No	Acc: 96.94
Mahjoubi et al. [45]	CNN	Softmax	7022 images. 4 classes (glioma, meningioma, pituitary, no tumor)	80:20	No	Acc: 95.44 Rec: 95.00 F1: 95.36
Mahmud et al. [46]	CNN	Softmax	3264 images. 3 classes (glioma, meningioma, pituitary)	80:10:10	Yes	Acc: 93.30 Rec: 91.19
Kumar et al. [47]	ResNet50	Softmax	1731 images. 2 classes (brain tumor, no-tumor)	90:10	No	Acc: 96.80 Pre: 94.50 Rec: 92.32 F1: 92.12
Abdusalomov et al. [48]	CNN	Softmax	10288 images. 4 classes (glioma, meningioma, pituitary, no tumor)	80:20	Yes	Acc: 99.50 F1: 99.40
Ozkaraca et al. [49]	CNN	Softmax	7021 images. 4 classes (glioma, meningioma, pituitary, no tumor)	80:20	No	Pre: 96.00 Rec: 96.50 F1: 96.00
Sharma et al. [50]	Modified ResNet50	Softmax	50 images. 2 classes (brain tumor, no-tumor)	5-fold CV	Yes	Acc: 88.00 Pre: 80.00 Spe: 100.0 Sen: 100.0
Asif et al. [51]	Xception + deep dense block	Softmax	Dataset A 3064 images. 3 classes (glioma, meningioma, pituitary) Dataset B 3264 images. 4 classes (glioma, meningioma, pituitary, no tumor)	80:20	Yes	Dataset A Acc: 99.67 Pre: 99.69 Spe: 99.83 Sen: 99.54 F1: 99.62

(*continued on next page*)

Table 1 (*continued*)

Study	Method	Classifier	Dataset / Number of classes	Split ratio	Augmentation	The results (%)
						Dataset B Acc: 95.87 Pre: 96.25 Spe: 98.56 Sen: 95.60 F1: 95.92
Kanchanamala et al. [52]	Exponential deer hunting optimization-based Shepard CNN	Softmax	3064 images. 3 classes (glioma, meningioma, pituitary)	80:20	Yes	Acc: 91.70 Spe: 91.90 Sen: 91.80
Muezzinoglu et al. [53]	PatchResNet	kNN	3264 images. 4 classes (glioma, meningioma, pituitary, no tumor)	10-fold CV	No	Acc: 98.10
Poyraz et al. [54]	Pyramid and fixed-size patch-histogram-oriented gradients, iterative Chi2, kNN	kNN	3264 images. 4 classes (glioma, meningioma, pituitary, no tumor)	10-fold CV	No	Acc: 94.67 Pre: 94.41 Rec: 93.84 F1: 94.10
Woźniak et al. [55]	Deep neural network correlation learning	Artificial neural network	3064 images. 3 classes (glioma, meningioma, pituitary)	83:17	Yes	Acc: 97.50 Sen: 95.60 Spe: 96.21
Khaliki et al. [56]	VGG16	Softmax	3264 images. 4 classes (glioma, meningioma tuor, pituitary, no tumor)	75:10:15	No	Acc: 98.00 Pre: 98.00 Rec: 98.00 F1: 97.00
Ullah et al. [29]	Deep explainable brain tumor deep network	Softmax	3000 images. 2 classes (brain tumor, no-tumor)	90:10	No	Acc: 94.00

following major novelties:

- Our model introduces FlexiViT-inspired multiple fixed-size patches (30 × 30, 35 × 35, 42 × 42, and 70 × 70) into traditional feature engineering, enabling efficient local feature extraction from different image regions.
- This research utilizes LBP, LPQ, and PHOG feature extractors in a unique combinatorial approach, creating a new-generation feature extraction model that leverages both textural and directional characteristics across different patch sizes.
- The presented FlexiCombFE incorporates self-organizing mechanisms, through iterative majority voting and greedy algorithm-based selection, enabling automatic optimization of feature selection and classification processes.

The practical impact and scientific contributions of our work can be summarized as follows:

- In this research, a self-organized feature engineering framework has been presented, inspired by deep learning models. In this regard, the introduced FlexiCombFE offers a competitive alternative to deep learning by utilizing traditional feature extraction functions.
- The FlexiCombFE achieved a 99.35 % classification accuracy on a public brain tumor dataset due to its self-organized structure and effectiveness. The modular design of the proposed FlexiCombFE opens new possibilities, similar to deep learning models, as various feature extraction, feature selection, and classification methods can be integrated into the framework. This allows for the development of many new-generation flexible and combinational feature engineering models.

2. Brain image dataset

In this study, we utilized a publicly accessible dataset [40] comprising two categories: (i) brain tumor and (ii) control. The dataset includes magnetic resonance (MR) images of varying dimensions, which we resized to a uniform resolution of 210 × 210 pixels. Additionally, the dataset encompasses images in various file formats, including.jpg,.tif, and.png. The dataset is available for download from the following URL: https://www.kaggle.com/datasets/preetviradiya/brian-tumor-dataset. The distribution of the dataset is illustrated in Table 2. Fig. 1 shows some sample images from this dataset for each class.

3. The FlexiCombFE

In this research, we have presented a new feature engineering model called FlexiCombFE. Our proposed framework contains four main phases, and these phases are:

- *Feature extraction:* In this phase, the images have been converted to grayscale and resized to 210 × 210 pixels. To generate features, we have used four types of fixed-size patches (30 × 30, 35 × 35, 42 × 42 and 70 × 70 pixels) and three feature extractors (LBP, LPQ, and PHOG). This phase generates 28 feature vectors.
- *Feature selection:* In this phase, we have used three feature selectors: NCA, Chi2, and RF. By deploying these feature selectors, the best 256 features were selected from each feature vector, resulting in 84 selected feature vectors.

Table 2
The distribution of the used brain image dataset.

Class	Number of images
Brain tumor	2513
Control	2087
Total	*4600*

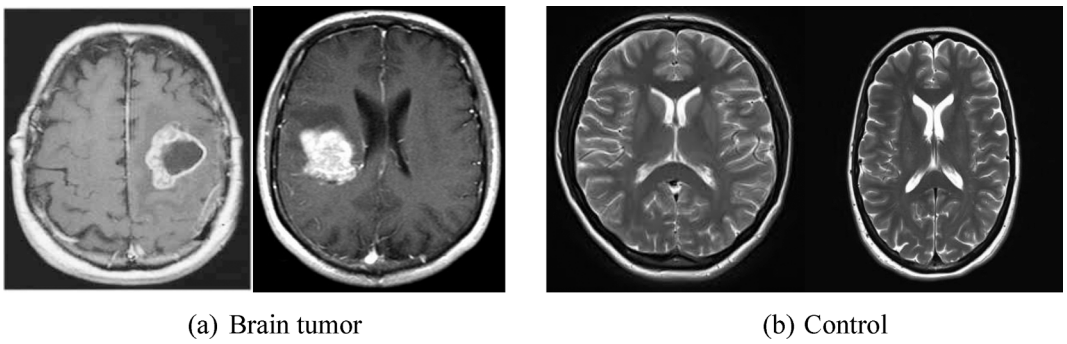

Fig. 1. Sample images of the used dataset.

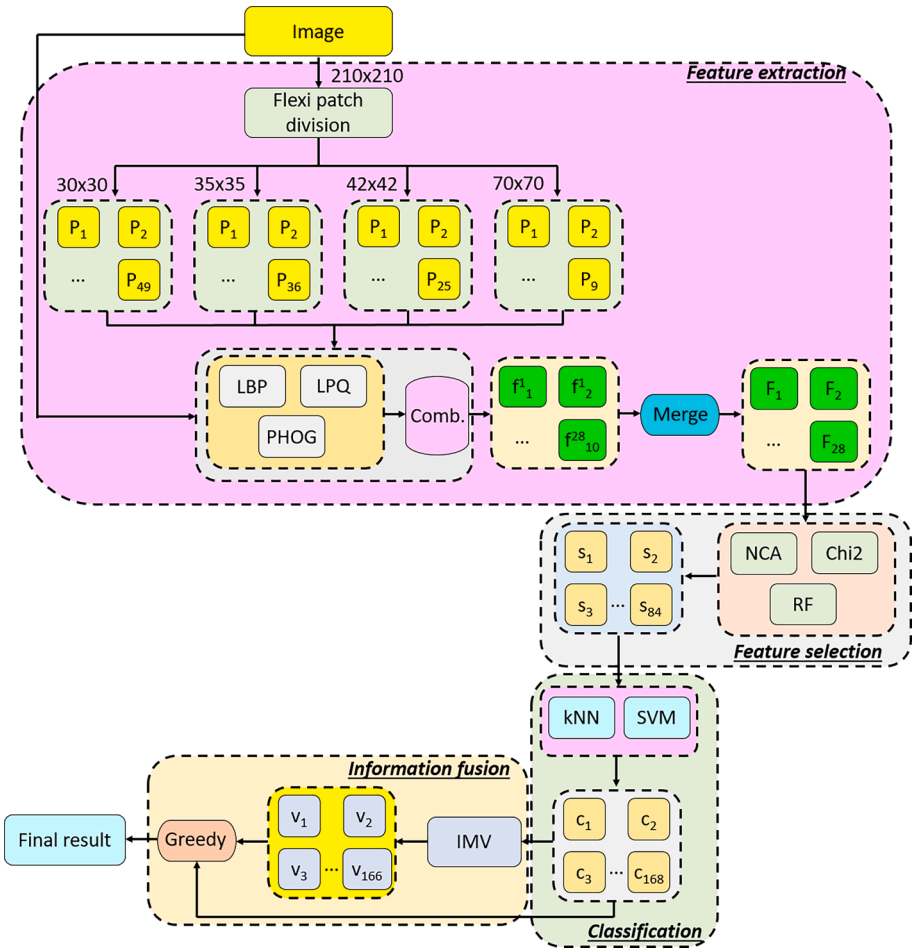

Fig. 2. Overview of the proposed FlexiCombFE. Here, P: patch, f: individual feature vector, LBP: local binary pattern, LPQ: local phase quantization, PHOG: pyramidal histogram oriented gradient, F: merged feature vector, NCA: neighborhood component analysis, Chi2: Chi-squared, RF: ReliefF, s: selected feature vector kNN: k-nearest neighbors, SVM: support vector machine, c: classifier-specific outcome, IMV: iterative majority voting, v: voted outcome.

- *Classification:* This phase has used two classifiers (kNN and SVM). The selected feature vectors in the feature selection phase have been utilized as input for these classifiers. In this phase, 168 classifier-based outcomes have been generated.
- *Information fusion:* The IMV algorithm has been used to generate 166 voted outcomes, resulting in a total of 334 outcomes. Then, a greedy algorithm has selected the most accurate outcome.

Fig. 2 presents a graphical outline of the proposed FlexiCombFE model to provide a better explanation of its structure and functionality. In the next subsection, a detailed description of the presented FlexiCombFE is provided.

3.1. Feature extraction

The initial phase of the proposed framework focuses on feature extraction, during which 28 feature vectors are generated. Fig. 3 demonstrates the presented feature extraction method.

The specifics of this phase are detailed as follows:

Step 1: Conversion of each image to grayscale and resize to 210×210 pixels.

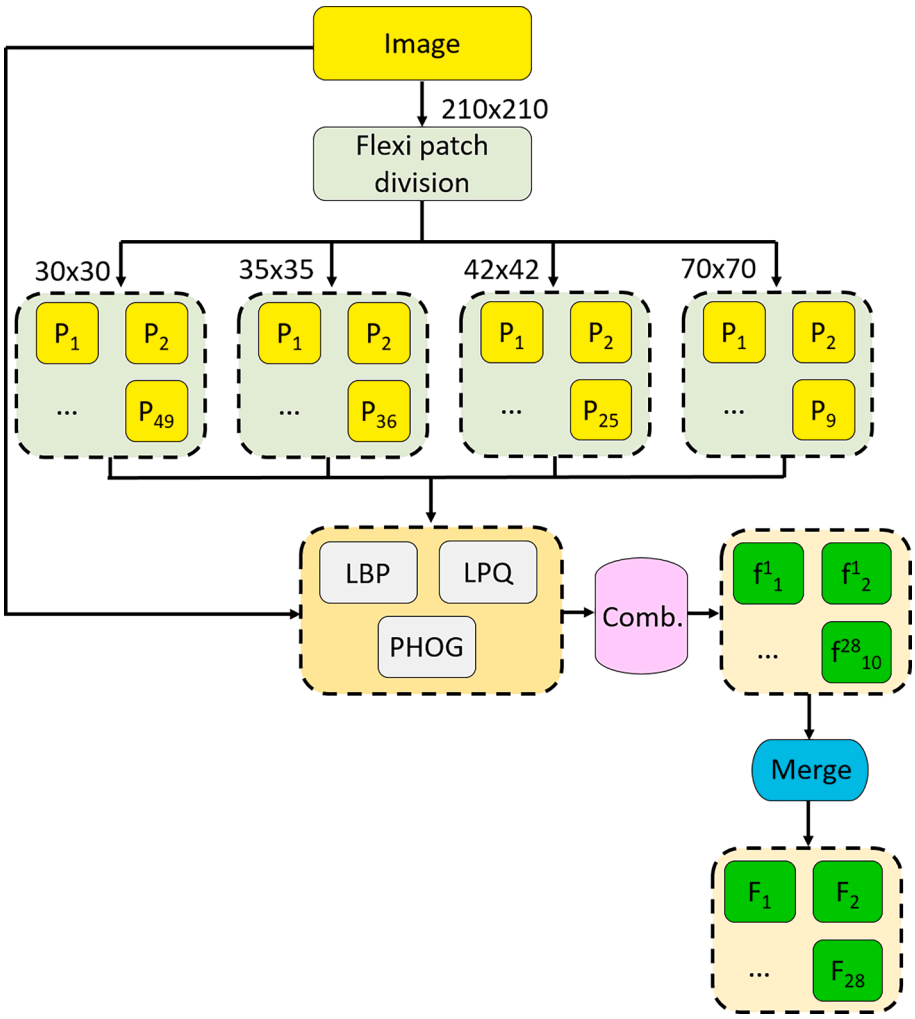

Fig. 3. Overview of the feature extraction method. The process used four different fixed-size patch division to extract local features from different image regions. Three feature extractors (LBP, LPQ and PHOG) are applied to these patches and the original image, generating 28 feature vectors through various combinations.

$$\begin{aligned} Gray &= RGB2Gray(Im) \\ Gray &= R(Gray, 210 \times 210) \end{aligned} \tag{1}$$

where *Im*: image, *Gray*: grayscale image, *RGB2Gray*(.): grayscale transformation function and *R*(.): resizing function. We have used 210 × 210 for resizing since 210 is the multiplication of the first four prime number (210 = 2 × 3 × 5 × 7).

Step 2: Generation of fixed-size patches using 30 × 30, 35 × 35, 42 × 42 and 70 × 70 pixel sizes.

$$\begin{aligned} &P_j^t = Gray(m : m + win(t) - 1, n : n + win(t) - 1), t \in \{1,2,3,4\}, \\ &win \in \{30,35,42,70\}, m,n \in \{1, win(t)+1, \cdots, 210 - win(t)\}, \\ &j \in \{1,2,\cdots,\frac{210}{win(t)}^2\} \end{aligned} \tag{2}$$

here, *P*: fixed-size patch. We have generated for types patches, with the number of these patches being 49, 36, 25, and 9, respectively.

Step 3: Feature extraction from the generated patches and raw images and patches to create individual feature vectors. We used LBP, LPQ, and PHOG feature extractors in this step. LBP and LPQ generate textural features, while PHOG generates directional features. We have created 7 (=2^3-1) feature vectors for each input by deploying combinations of these feature extractors. First, we have defined the presented combination-based feature extractor.

$$\begin{aligned} feat_1 &= LBP(In), \\ feat_2 &= LPQ(In), \\ feat_3 &= PHOG(In), \\ feat_4 &= concat(feat_1, feat_2) \\ feat_5 &= concat(feat_1, feat_3) \\ feat_6 &= concat(feat_2, feat_3) \\ feat_7 &= concat(feat_1, feat_2, feat_3) \end{aligned} \tag{3}$$

where *In*: input, *feat*: the generated feature vectors, and *concat*(.): the concatenation function. Using this combination feature extractor, we have generated seven feature vectors for each input. We have applied this feature extraction function to the generated patches and raw images to create individual feature vectors.

$$\begin{aligned} [f_1^{4(t-1)+1}, f_1^{4(t-1)+2}, \cdots, f_1^{4(t-1)+7}] &= CombFE(Gray) \\ [f_{j+1}^{4(t-1)+1}, f_{j+1}^{4(t-1)+2}, \cdots, f_{j+1}^{4(t-1)+7}] &= CombFE(P_j^t) \end{aligned} \tag{4}$$

Then, we have extracted features from both raw images and patches. In this case, we have generated 50 (=49 + 1), 37 (=36 + 1), 26 (=25 + 1) and 10 (=9 + 1) features using the 30 × 30, 35 × 35, 42 × 42 and 70 × 70 sized patches respectively.

Step 4: Merge the features according to feature types and generate 28 (= 7 feature vectors × 4 patches) merged feature vectors.

$$F_q = concat\left(f_1^q, f_2^q, \cdots, f_{j+1}^q\right), q \in \{1,2,\cdots,28\}, \tag{5}$$

where F defines the merged feature vector. To clarify Step 1–4, we have provided the pseudocode of the proposed feature extraction methodology in Algorithm 1. By deploying these steps (Algorithm 1 explained these steps), our model generates 28 feature vectors.

Algorithm 1 Pseudocode of the presented combination-based flexible feature extraction.

Input: Image

Output: Merged feature vectors
01: Apply preprocessing by deploying Eq. (1).
02: $win \in \{30, 35, 42, 70\}$;
03: $c = 1$; // Counter definition
04: **for** t = 1 to *length*(*win*) **do**
05: $[F_c(1:l_1), F_{c+1}(1:l_2), \cdots, F_{c+6}(1:l_7)] = CombFE(Gray)$
06: $c_1 = 1$; // The second counter definition for feature concatenation
07: **for** w = 1 to 210 step by $win(t)$ **do**
08: **for** h = 1 to 210 step by $win(t)$ **do**
09: $patch = Gray(w : w + win(t) - 1, h : h + win(t) - 1)$;
10: $F_c(c_1 * l_1 + 1 : (c_1 + 1) * l_1), F_{c+1}(c_1 * l_1 + 1 : (c_1 + 1) * l_1), \cdots$
$F_{c+6}(c_1 * l_7 + 1 : (c_1 + 1) * l_7)] = CombFE(patch)$;
// Herein, feature extraction and concatenation stages are merged.
11: $c_1 = c_1 + 1$;
12: **end for h**
13: **end for w**
14: $c = c + 7$;
15: **end for t**

3.2. Feature selection

In this phase, we utilized NCA, Chi2, and RF as feature selectors to generate 84 selected feature vectors from the initially created 28 feature vectors. The objective of this phase was twofold: to augment the number of feature vectors and to standardize their lengths. Consequently, we selected the top 256 features from each of the 28 generated feature vectors. The schematic diagram of the presented feature selection is depicted in Fig. 4.

The procedural steps of the feature selection phase are as follows:

Step 5: Creation of the qualified indices of the feature vectors by utilizing NCA, Chi2, and RF feature selectors.

$$\begin{aligned} idx_q &= NCA(F_q, y), \\ idx_{q+28} &= Chi2(F_q, y), \\ idx_{q+56} &= RF(F_q, y) \end{aligned} \tag{6}$$

where *idx*: the qualified indices (84 qualified indices have been created) and *y*: actual/real output.

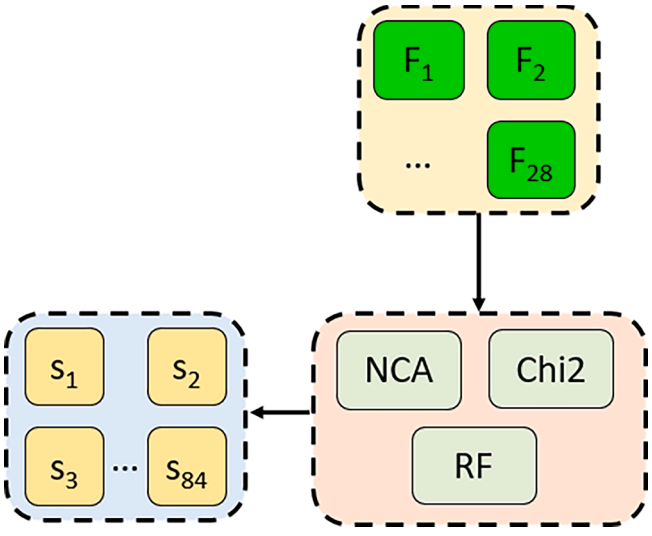

Fig. 4. Feature selection process overview. Three feature selectors (NCA, Chi2, and ReliefF) are applied to the 28 generated feature vectors, producing 84 selected feature vectors (28 × 3).

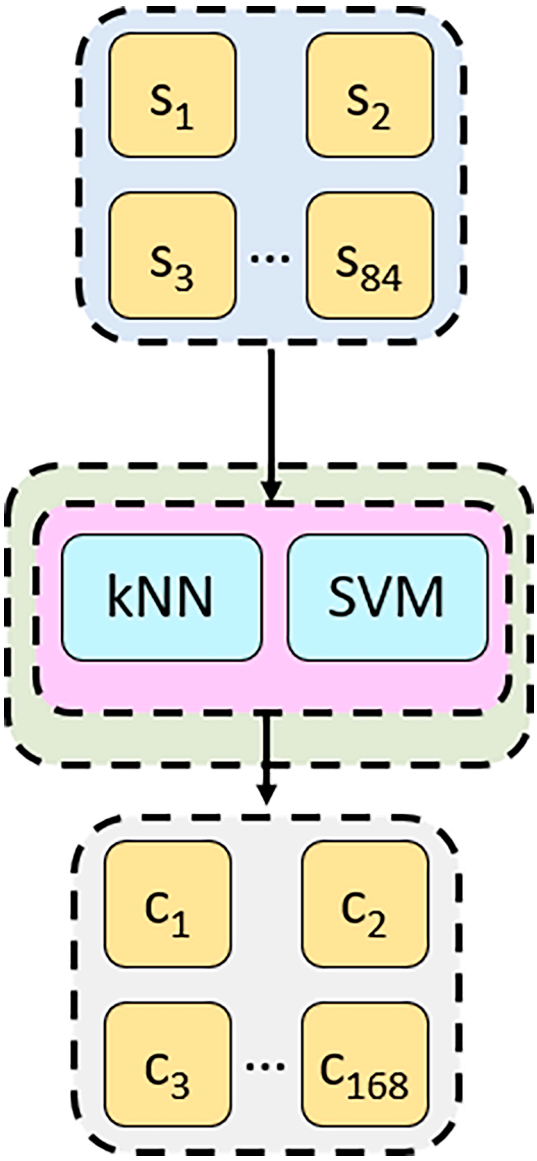

Fig. 5. Classification phase workflow. The 84 selected feature vectors are processed using kNN and SVM classifiers, generating 168 (84 × 2) outcomes.

Step 6: Definition of the qualified indices of the feature vectors by utilizing NCA, Chi2, and RF feature selectors.

$$\begin{aligned} s_q(d,g) = F_q(d, idx_q(g)), g \in \{1,2,\cdots,256\}, d \in \{1,2,\cdots,NI\} \\ s_{q+28}(d,g) = F_q(d, idx_{q+28}(g)), \\ s_{q+56} = F_q(d, idx_{q+28}(g)) \end{aligned} \tag{7}$$

where *s*: selected feature vector and *NI*: number of images.

3.3. Classification

In this study, we aimed to demonstrate the high classification ability of the features generated by the proposed FlexiCombFE framework. To achieve this, we utilized two shallow classifiers, kNN and SVM. These classifiers were selected based on their performance within the MATLAB Classification Learner Toolbox. This toolbox provides over 30 traditional and ensemble classifiers for evaluation. After conducting an extensive performance analysis, kNN and SVM emerged as the top-performing classifiers for the given dataset. Hence, we chose these classifiers for our study. While we acknowledge that advanced deep learning models might offer competitive performance, our focus was to showcase the

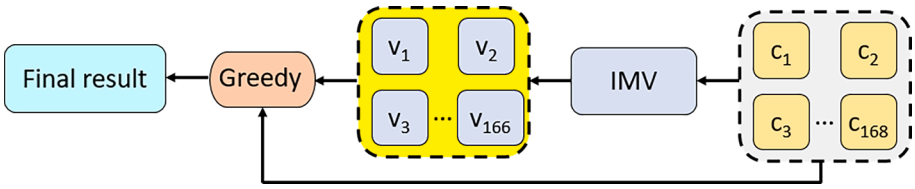

Fig. 6. Information fusion phase overview. The process takes 168 classifier-specific outcomes as input. IMV produces 166 additional voted outcomes, resulting in 334 total outcomes. The final step selects the best outcome among these.

strength of the feature engineering approach. The classification phase of the introduced Fig. 5.

The SVM and kNN classifiers were applied to the selected feature vectors as inputs, generating 168 classifier-specific outcomes. The steps involved in the classification phase are as follows:

Step 7: Creation of the qualified indices of the feature vectors by utilizing NCA, Chi2, and RF feature selectors.

$$\begin{aligned} c_p &= kNN(s_p, y), p \in \{1, 2, \cdots, 84\} \\ c_{p+84} &= SVM(s_{p+84}, y), \end{aligned} \tag{8}$$

where c: classifier-specific outcome.

3.4. Information fusion

The IMV method was used to produce the voted result by combining the strengths of individual classifiers using an iterative majority voting mechanism. This approach ensures that the collective decision utilizes reliable results, thereby reducing the effect of poorly performing classifier-based outcomes. Following this, a greedy algorithm was applied to determine the single best classification result from the obtained outcomes. The greedy algorithm allowed us to achieve the optimum outcome by selecting the result with the highest accuracy and identifying the best combination of feature extraction, feature selection, and classification models. The graphical depiction of the introduced information fusion has been depicted in Fig. 6.

The IMV method and a greedy algorithm were applied in the final phase. Utilizing IMV, we produced voted outcomes based on the 168 classifier-specific outcomes, resulting in 166 voted outcomes. The steps of this phase are detailed below:

Step 8: Generation of the voted outcomes by deploying IMV and classifier-specific outcomes.

$$\begin{aligned} ac &= \psi(c_a, y), a \in \{1, 2, \cdots, 168\}, \\ ix &= Sort(ac), \\ v_b &= \omega(c_{ix(1)}, c_{ix(2)}, \cdots, c_{ix(b+2)}), b\{1, 2, \cdots, 166\} \end{aligned} \tag{9}$$

where ac: classification accuracy, $\psi(.)$: classification accuracy calculation function, ix: the qualified indices of the classifier-specific outcomes per classification accuracy by descending, $Sort(.)$: sorting function, v: voted outcome and $\omega(.)$: mode function.

Step 9: Selection of the best outcome as the ultimate outcome.

$$\begin{aligned} ac(a) &= \psi(c_a, y), \\ ac(168+n) &= \psi(v_n, y), n \in \{1, 2, \cdots, 166\}, \\ [maxi, x] &= max(acc), \\ ult &= \begin{cases} c_x, x \leq 168 \\ v_{x-168}, x > 168 \end{cases} \end{aligned} \tag{10}$$

where ac: classification accuracy, $maxi$: the maximum classification accuracy, x: index of the maximum classification accuracy, and ult: ultimate outcome.

In this phase, Step 8 delineates the implementation of the IMV algorithm, while Step 9 outlines the application of the greedy algorithm. These steps collectively constitute the methodology of the proposed FlexiCombFE framework.

4. Experimental results

In this section, the performance evaluation results of the presented FlexiCombFE have been presented and the details of these results are given below.

4.1. Time complexity of the FlexiCombFE model

The FlexiCombFE is a lightweight model based on feature engineering. This model consists of four main phases, and their time complexity computations using Big O notation are provided below.

Feature Extraction: The feature extraction phase employs fixed-size patch with different combinations. We have employed simple feature extraction functions: (i) LBP, (ii) LPQ, and (iii) PHOG. The time complexity of these functions is $O(S)$, where S represents the size of the patch. Therefore, the time complexity of the feature extraction phase is $O(3SP + C) = O(SP + C)$, where P defines the number of patches and C represents the time complexity of the combination process.

Feature Selection: Three different feature selectors have been used in this phase: NCA, ReliefF, and Chi2. The time complexity is $O(RF + NF + CF)$, where R, N, and C are the time complexity coefficients of ReliefF, NCA, and Chi2, respectively, and F represents the number of features.

Classification: Two shallow classifiers, kNN and SVM, are used to generate classification outputs. The time complexity of this phase is $O(TK + TV)$, where T represents the number of selected feature vectors, K is the time complexity coefficient of kNN, and V is the time complexity coefficient of SVM.

Information Fusion: This phase uses IMV and the greedy algorithm to choose the best classification outcome. The time complexity is $O(I + G)$, where I represents the time complexity coefficient of IMV and G represents the time complexity of the greedy algorithm.

Total: The total time complexity of the FlexiCombFE is $O(SP + C + F(R + N + C) + T(K + V) + I + G)$. This result demonstrates the linear time complexity of the proposed model.

4.2. Classification performance of the FlexiCombFE model

In this section, we detail the classification performance of the FlexiCombFE framework. The FlexiCombFE framework was developed within the MATLAB (2023a) programming environment. For classification, the selected features were processed using the Classification Learner Toolbox (MATLAB version 2024a). We chose the two most effective shallow classifiers: kNN and SVM.

The FlexiCombFE framework was executed on a personal computer equipped with 32 GB of RAM and a 3.2 GHz processor, and it ran the Windows 11 operating system. The implementation was carried out in CPU mode, highlighting the framework's nature as a hand-crafted model designed for high-performance computational tasks. The transition table of the proposed model, detailing the process flow and outcomes, is presented in Table 3.

To assess the performance of the proposed FlexiCombFE framework, we employed a 10-fold cross-validation strategy. We utilized a comprehensive set of performance metrics, including classification

Table 3
Transition table of the presented FlexiCombFE.

Phase	Methods	Parameters	Output
Feature extraction	Grayscale conversion	Input: RGB image, Multipliers:R: 0.2989, G: 0.5870 and B: 0.1140	Grayscale image
	Image resizing	Bilinear interpolation	210 × 210 sized image
	Patch division	Patch sizes: 30 × 30, 35 × 35, 42 × 42 and 70 × 70.	Number of patches: For 30 × 30: 49, For 35 × 35: 36, For 42 × 42: 25, For 70 × 70: 9.
	Feature extraction deploying LBP, LPQ, PHOG and combinations of them	Feature extractor 1: LBP, Feature extractor 2: LPQ, Feature extractor 3: PHOG, Feature extractor 4: LBP + LPQ, Feature extractor 5: BLP + PHOG, Feature extractor 6: LPQ + PHOG, Feature extractor 7: LBP + LPQ + PHOG	Number of features: Feature vector 1: 59, Feature vector 2: 256, Feature vector 3: 168, Feature vector 4: 315, Feature vector 5: 227, Feature vector 6: 424,Feature vector 7: 483,
	Generation 28 feature vectors	Feature extraction from raw image and the generated patches. Merge the features per the used patch and feature extractors	Number of features: For 30 × 30: *1:* 2950, *2:* 12800, *3:* 8400, *4:* 15750, *5:* 11350, *6:* 21200, *7:* 24150. For 35 × 35: *8:* 2183, *9:* 9472, *10:* 6216, *11:* 11655, *12:* 8399, *13:* 15688, *14:* 17871. For 42 × 42: *15:* 1534, *16:* 6656, *17:* 4368, *18:* 8190, *19:* 5902, *20:* 11024, *21:* 12558. For 70 × 70: *22:* 590, *23:* 2560, *24:* 1680, *25:* 3150, *26:* 2270, *27:* 4240, *28:* 4830.
Feature selection	Feature selection with NCA, Chi2 and RF	Input: Generated 28 feature vectors. Three feature selectors were used with default setting.	The lengths of the selected feature vectors 256. Selected feature vectors 1–28 have been generated using NCA. Selected feature vectors 29–56 have been generated deploying Chi2. Selected feature vectors 57–84 have been generated utilizing RF.
Classification	Classification with kNN and SVM	Input: 84 selected feature vectors. kNN: k:1, distance: L1-norm, voting: none, validation: 10-fold cross-validation. SVM: Kernel: 3rd degree polynomial, box constraint: 1, coding: one-vs-one, validation: 10-fold cross-validation.	Outcomes 1–84 have been created by the kNN. Outcomes 85–168 have been produced using SVM.
Information fusion	Voted outcomes generation with IMV	Range of iteration: from 3 to 168, Sorting criteria: Accuracy by descending, Voting function: Mode.	The number of generated voted outcomes: 166
	Greedy algorithm	Input: 334 outcomesSelection criteria: Maximum accuracy	The best outcome

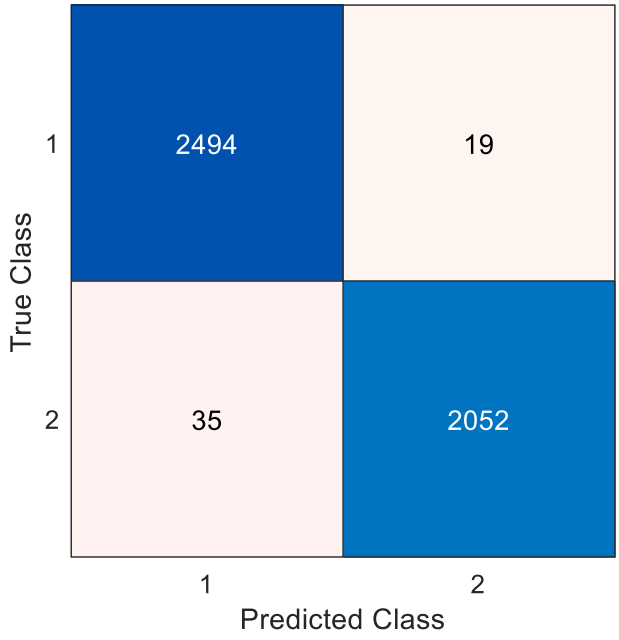

Fig. 7. Confusion matrix of the best classifier-based outcome. 1: Brain Tumor, 2: Control. Blue cells indicate correctly predicted observations, while beige cells show incorrect predictions. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 4
The computed classification performances of the best classifier-specific outcome of the FlexiCombFE.

Performance metric	Result (%)
Accuracy	98.83
Sensitivity	99.24
Specificity	97.51
Geometric mean	98.37
Precision	98.62
F1-score	98.93

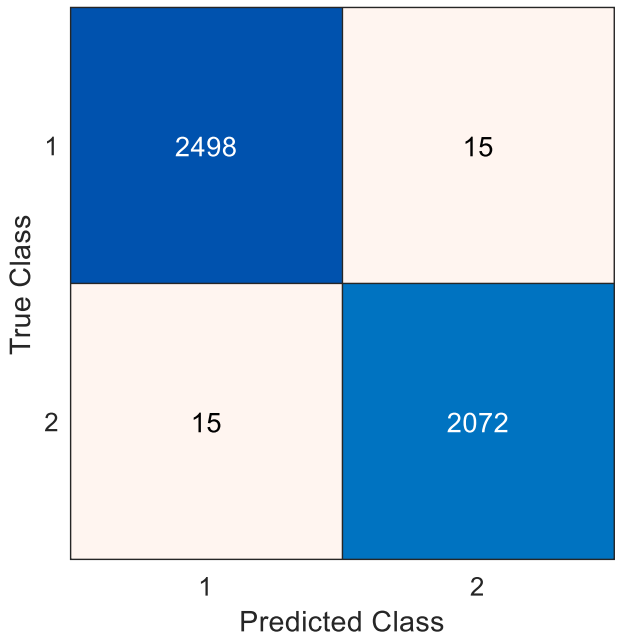

Fig. 8. Confusion matrix of the final outcome. Here, 1: Brain Tumor, 2: Control. Blue cells indicate correctly predicted observations, while beige cells show incorrect predictions. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

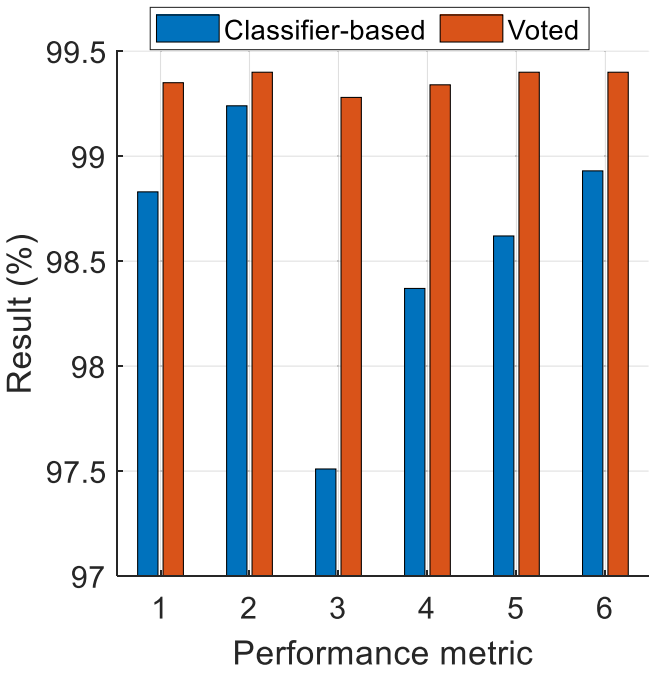

Fig. 9. Performance comparisons of the best classifier-based and voted outcomes. Herein, 1: Accuracy, 2: Sensitivity, 3: Specificity, 4: Geometric mean, 5: Precision, 6: F1-score.

Table 5
The computed classification performance of the proposed FlexiCombFE.

Performance metric	Result (%)
Accuracy	99.35
Sensitivity	99.40
Specificity	99.28
Geometric mean	99.34
Precision	99.40
F1-score	99.40

accuracy, sensitivity, specificity, geometric mean, precision, and F1-score.

The FlexiCombFE framework produced outcomes on an individual classifier basis and through a voting mechanism, with the ultimate decision derived from the voted outcome. The most effective classifier-based outcome was identified as the 27th outcome, generated according to the following configuration: the feature extraction method combined LPQ with PHOG (LPQ + PHOG), the feature selection was conducted using NCA, and kNN performed the classification. The confusion matrix for this optimal classifier-based outcome, depicted in Fig. 7, served as the basis for calculating the classification performance metrics. Based on the confusion matrix results, the framework achieved an impressive classification accuracy of 98.83 %, with only 54 misclassified cases out of 4600 total images.

Utilizing the data from Fig. 7, the classification performance metrics for the optimal classifier-based outcome are systematically compiled in Table 4.

Table 4 reveals that the optimal classifier-based outcome achieved an accuracy of 98.83 %, a geometric mean of 98.37 %, and an F1-score of 98.93 %. The final result from the FlexiCombFE framework is derived from a voted outcome. To calculate the performance metrics of this ultimate outcome, we referred to the confusion matrix corresponding to the ultimate output, as presented in Fig. 8. The final outcome demonstrates superior performance with an accuracy of 99.35 %, showing significant improvement through information fusion with only 30 misclassified cases out of 4,600 total images.

According to Fig. 8, the performance metrics calculated for the final outcome are summarized in Table 5. This table shows that the proposed FlexiCombFE model achieved a classification accuracy of 99.35 %, a geometric mean of 99.34 %, and an F1-score of 99.40 %. This ultimate outcome, identified as the 13th-voted outcome, was derived from voting among the top 15 classifier-specific outcomes.

A comparison of the results in Table 4 and Table 5 demonstrates the effectiveness of the information fusion phase in enhancing the classification performance of the proposed model. Fig. 9 further illustrates these improvements by comparing the performance metrics before and after the information fusion phase. As shown in Fig. 9, the application of information fusion significantly enhanced all classification performance metrics of the proposed model. This improvement underscores the effectiveness of the information fusion phase in boosting the model's overall accuracy and reliability.

4.3. Comparative analysis of model components

We have evaluated the classification performance of the feature extractors, feature selectors, and classifiers based on the 168 classifier-specific outcomes. We used box plot analysis to present these methods' statistical properties graphically. The graphical results for the utilized methods are illustrated in Fig. 10.

These analyses were conducted using classifier-specific results, with an evaluation of the employed methods based on average classification accuracy. The key findings from this analysis are:

- PHOG was identified as the most effective feature extractor, achieving an average classification accuracy of 98.12 %. However, the highest classifier-specific outcome was produced using a combination of the LPQ and PHOG feature extractors.
- The optimal patch size was identified as 70 × 70, which yielded an average classification accuracy of 98.02 %.
- The most effective feature selector was NCA, which achieved an average classification accuracy of 98.01 %.
- In terms of classifiers, the average classification accuracies for kNN-based and SVM-based outcomes were 98.15 % and 97.43 %,

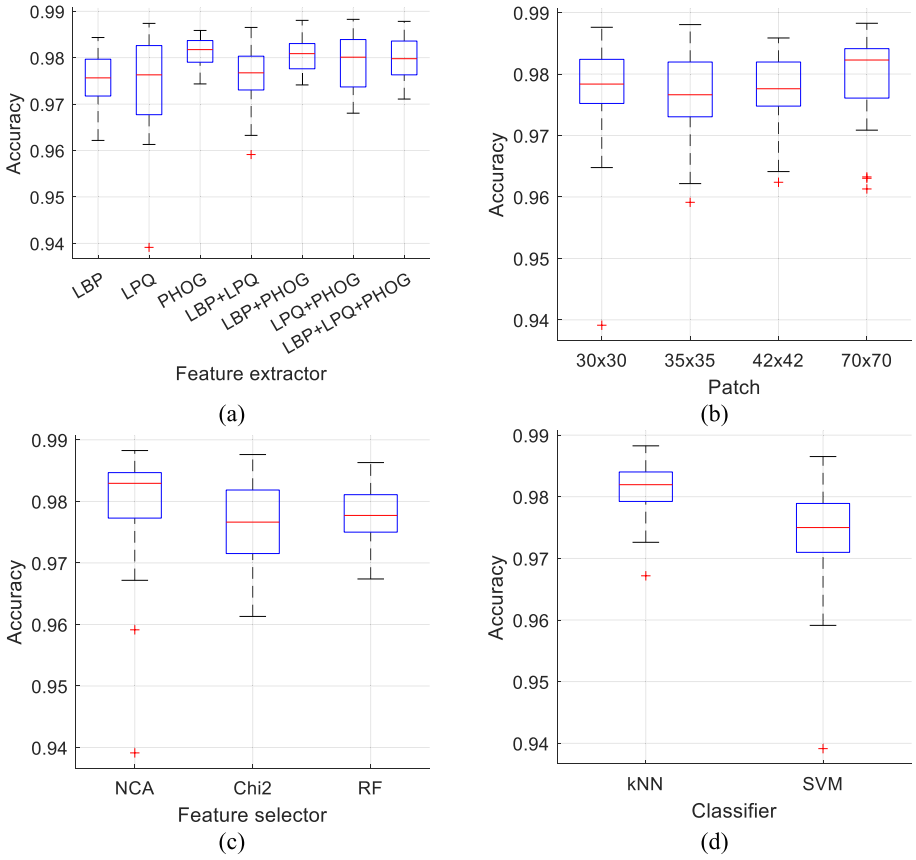

Fig. 10. Evaluation of the used methods.

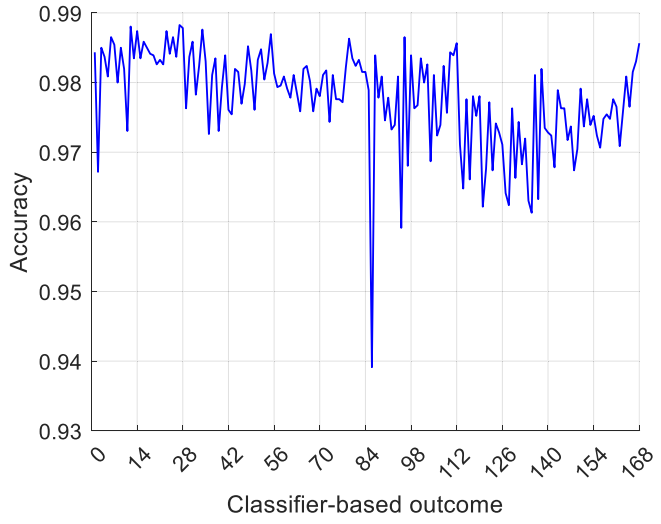

Fig. 11. The classification accuracies of the classifier-based outcomes.

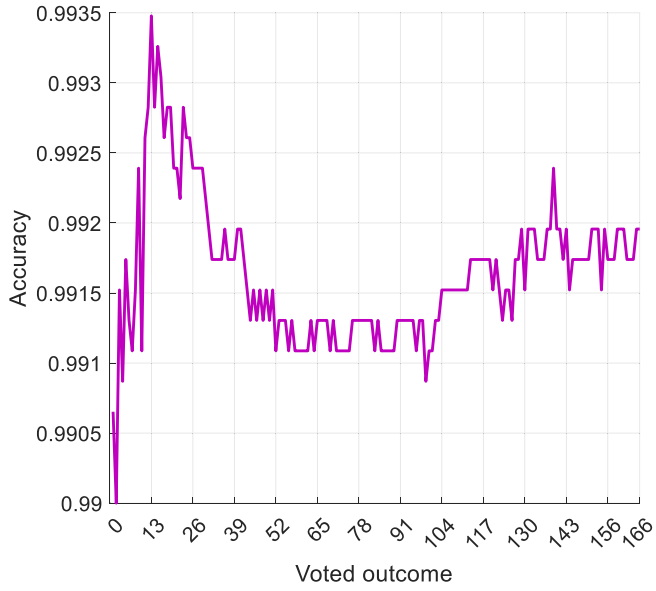

Fig. 12. The classification accuracies of the generated 166 voted outcomes.

respectively, indicating that the kNN classifier outperformed the SVM classifier.

In addition to the above analysis, the classification accuracies of the computed classifier-specific outcomes are depicted in Fig. 11.

According to Fig. 11, the optimal outcome was identified as the 27th, achieving an accuracy of 98.83 %. This result was produced using a combination of the LPQ + PHOG feature extractor with a 70 × 70 patch division, the NCA feature selector, and the kNN classifier. Conversely, the least accurate outcome was the 86th (which reached 93.91 % classification accuracy), generated with the LBP feature extractor using a 30 × 30 patch division, the NCA feature selector, and the SVM classifier. Notably, the NCA feature selector was involved in both the highest and lowest classifier-specific outcomes.

With the application of IMV, the voted outcomes within the FlexiCombFE framework were generated, showcasing classification accuracies of 99 % or higher. The classification accuracies of these voted outcomes are illustrated in Fig. 12.

Table 6
The used classifier-based outcomes to generate ultimate outcome.

No	Feature extractor	Patch size	Feature selector	Classifier
27	LPQ + PHOG	70 × 70	NCA	kNN
12	LBP + PHOG	35 × 35	NCA	kNN
28	LBP + LPQ + PHOG	70 × 70	NCA	kNN
34	LPQ + PHOG	30 × 30	Chi2	kNN
14	LBP + LPQ + PHOG	35 × 35	NCA	kNN
23	LPQ	70 × 70	NCA	kNN
55	LPQ + PHOG	70 × 70	Chi2	kNN
6	LBP + PHOG	30 × 30	NCA	kNN
25	LBP + LPQ	70 × 70	NCA	kNN
96	LBP + PHOG	35 × 35	NCA	SVM
79	LPQ	70 × 70	RF	kNN
16	LPQ	42 × 42	NCA	kNN
31	PHOG	30 × 30	Chi2	kNN
112	LBP + LPQ + PHOG	70 × 70	NCA	SVM
168	LBP + LPQ + PHOG	70 × 70	RF	SVM

The data presented in Fig. 12 shows that the 13th-voted outcome represents the outcome. The top 15 classifier-based outcomes (=13 + 3–1) were utilized to generate this particular outcome. The composition of this ultimate outcome was scrutinized using a counting-based analysis. Through an analysis of the methodologies listed in Table 6, our objective was to identify the most frequently utilized methods contributing to the generation of the ultimate result. This analysis, based on histogram representations, facilitates the identification of 'hotspot' methods. The histograms derived from this analysis are illustrated in Fig. 13.

Fig. 13. The frequency analysis of the used method to create ultimate outcome. Based on the analysis presented in Fig. 13, the following insights were identified:

- The top-performing feature extractor combinations are LPQ + PHOG and LBP + LPQ + PHOG. Furthermore, it has been observed that it is not necessary to incorporate the LBP feature extractor alone to generate the ultimate outcome.
- The optimal patch size for achieving the best results is determined to be 70 × 70.
- The NCA feature selector is superior to the Chi2 and RF feature selectors in contributing to the ultimate outcome.
- The kNN classifier demonstrates better performance than the SVM classifier.

4.4. Comparison with state-of-the-art

To illustrate the classification efficiency of the presented

Table 7
The comparative results with cutting-edge models.

Study	Method	Split ratio	Accuracy (%)
Bala et al. [57]	Local binary patterns	10-fold CV	99.12
Jain and Jaidka [58]	SVM + CNN	66:34	98.90
Chmiel et al. [59]	ResNet50	80:20	98.99
Valero Gómez [60]	MobileNet	80:20	98.59
Ads et al. [61]	Multi-limb split learning	80:20	90.69
Chen [62]	EfficentNet	67:33	91.00
Our study	The proposed FlexiCombFE	10-fold CV	99.35
	The best classifier-specific outcome of the FlexiCombFE		98.83

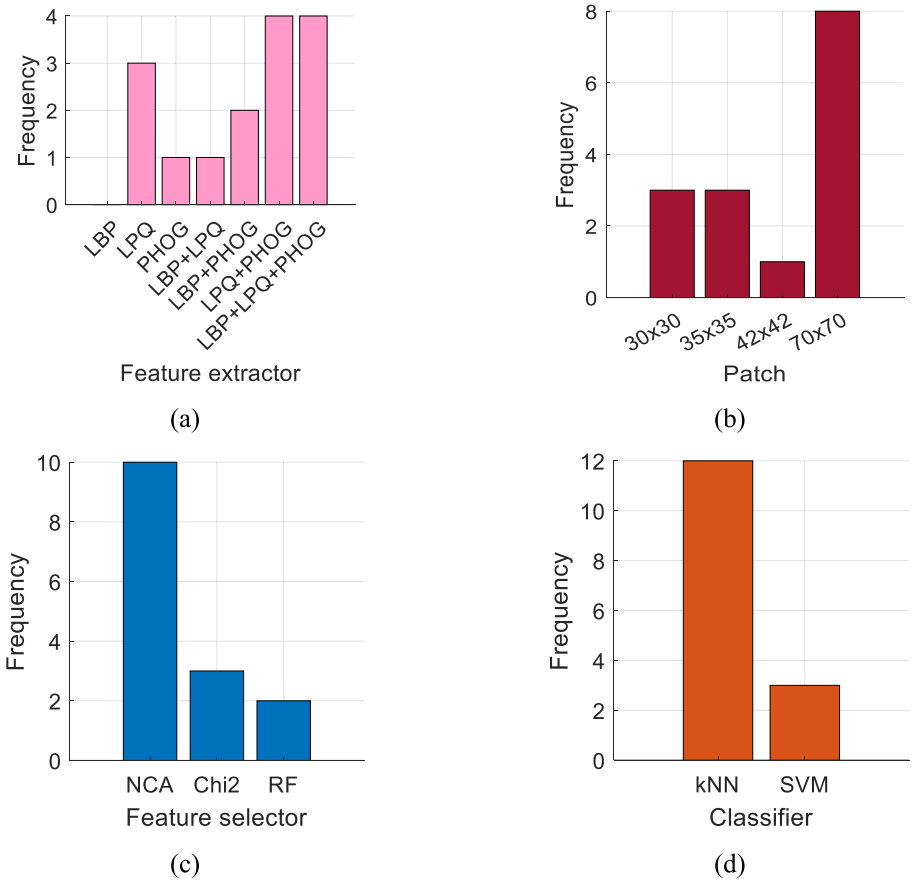

Fig. 13. Frequency analysis of the methods used to create ultimate outcome.

FlexiCombFE framework, we have conducted a comparison with state-of-the-art models. The comparative results are detailed in Table 7. According to Table 7, our proposed FlexiCombFE framework achieved higher classification accuracy than existing deep learning models. This achievement underscores our success in developing a competitive feature engineering model that can rival the performance of deep learning approaches.

5. Discussions

In this study, we introduced a novel framework named FlexiCombFE, designed to showcase its classification capabilities. We utilized an open-access (publicly available) biomedical image dataset comprising 4600 brain tumor images previously employed by deep learning models. Our main goal was to develop a hand-crafted (feature engineering) approach that achieves classification performance comparable to, or even surpassing, deep learning models.

The FlexiCombFE framework incorporates a unique combination of feature extraction techniques (LBP, LPQ, and PHOG), feature selection methods (NCA, Chi2, and RF), and classification algorithms (kNN and SVM). By using information fusion techniques (IMV and greedy algorithm), the presented model is converted to a self-organized form since it selected the best outcome among to the generated 334 outcomes.

One of the key strengths of FlexiCombFE is high classification accuracy and the introduced FlexiCombFE has often outperformed deep learning models. The high classification ability shows that the introduced FlexiCombFE can be used in the critical medical scenarios.

The information fusion technique based on IMV and greedy algorithm plays a crucial role in its performance. The IMV and greedy-based information fusion method combine classifier outcomes to obtain voted outcome and the voted outcomes have generally yielded higher classification performance than classifier-wise outcomes.

The utilized machine learning methods were analyzed, and the obtained findings are given as below. Among the evaluated feature extractors, PHOG achieved the highest average classification accuracy of 98.12 %, demonstrating its effectiveness in capturing directional and edge-related features essential for tumor distinction. Additionally, the combination of LPQ and PHOG highlights the importance of combining texture and directional features for comprehensive image analysis; the best classifier-wise result was obtained with this feature extraction combination (LPQ + PHOG). The optimal patch size was found to be 70 $\times$ 70 pixels, achieving an average accuracy of 98.02 %, as it effectively balances image detail and computational efficiency. In terms of model components, NCA proved to be the most effective feature selector, while the kNN classifier consistently outperformed the SVM classifier in classification tasks.

Another advantage of FlexiCombFE is its modular architecture. Unlike many deep learning models that operate as “black boxes,” FlexiCombFE maintains distinct feature extraction and classification stages. This transparency is particularly advantageous in applications demanding high interpretability, such as medical diagnosis and risk assessment, where understanding the model’s reasoning is paramount.

The introduced FlexiCombFE model tested achieved 99.35 % of classification accuracy and the used methods in this framework are analyzed in the previous section.

In the feature extraction phase, errors primarily arise from suboptimal patch sizes or inefficient feature extractors. To solve this problem, we diversified features using multiple patch sizes (30 $\times$ 30, 35 $\times$ 35, 42 $\times$ 42, and 70 $\times$ 70) and three complimentary feature extractors (LBP, LPQ, and PHOG). In addition, we generated multiple feature vectors using combination-based feature vector generation methods. Given that feature selection methods in the literature exhibit distinct selection capabilities, we employed three well-established feature selectors: NCA, Chi2, and ReliefF. These selectors not only reduce selection errors but also minimize errors in the generated feature vectors by eliminating redundant features.

For classification, we selected kNN and SVM classifiers based on their superior performance in the MATLAB Classification Learner tests. The implementation of both classifiers helps minimize classification errors through complementary decision-making approaches. In the information fusion stage, the IMV algorithm combines the classifier-specific results to reduce the impact of low performing models while minimizing errors propagation. The IMV generates robust outputs by laveraging the strengths of multiple classifiers, followed by a greedy algorithm that selects the most accurate result, thereby compensating for the errors introduced in the previous stages.

The modular architecture of FlexiCombFE facilitates error containment at each processing stage. The combination of multiple methods and IMV minimize error propagation and increase the reliability of the final classification result. Moreover, our experimental results (Figs. 10-13) demonstrate the individual classification performance of each method, confirming that the framework enhances the classification capabilities of the constituent models. This validates our approach of utilizing optimal configurations for each component.

Our research introduces a new image classification framework termed FlexiCombFE, consisting of four main phases: patch-based feature extraction, multiple selectors-based feature selection, kNN and SVM-based classification, and IMV and greedy-based information fusion. The framework achieved 99.35 % classification accuracy on a public brain tumor dataset, outperforming existing deep learning models. FlexiCombFE generates 28 feature vectors, 84 selected feature vectors, 168 classifier-specific outcomes, and 166 voted outcomes, making it self-organized by selecting the best outcomes from 334 generated results.

Analysis of the framework revealed several key findings: FlexiCombFE maintains linear time complexity as a lightweight model; the combination of LPQ + PHOG proved most effective for feature extraction; 70 $\times$ 70 pixels emerged as the optimal patch size; NCA showed superior performance in feature selection; and kNN consistently outperformed SVM. The modular structure of our framework enables integration of various feature engineering methods, while the IMV and greedy algorithm fusion phase enhanced overall accuracy.

The FlexiCombFE framework has several advantages that highlight its usefulness in applications related to medical imaging. The lightweight design allows for deployment on embedded devices, and its modular structure helps during the integration of various feature engineering techniques. Unlike typical “black-box” deep learning models, the system offers interpretability thanks to the use of traditional feature engineering techniques.

It is important to recognize that the FlexiCombFE architecture has some limitations despite its excellent performance. First, the framework’s efficacy is largely dependent on the careful selection of suitable feature extractors, classifiers, patch sizes, and feature selectors. These settings might need to be modified for other medical imaging applications, even if our experimental findings show the best combinations for brain tumor detection. Second, to show the framework’s wider application in clinical settings, more validation across various medical imaging datasets would be beneficial. These limitations offer chances for future studies to improve the framework’s versatility across a range of medical imaging applications. Future work will focus on the automation of parameter selection to present a fully automated version of FlexiCombFE. The framework exhibits potential for a number of medical applications, such as the detection of brain tumors in MRI or CT images, the categorization of cancer subtypes, the identification of lung conditions, the study of skin lesions, and the detection of cardiac abnormalities utilizing a variety of imaging modalities.

From a radiological perspective, FlexiCombFE offers valuable support to clinical practice. Its patch-based approach aligns with radiologists’ analytical methods, while its high accuracy and interpretability build trust in clinical settings. The lightweight design ensures accessibility in low-resource settings, and its versatility enables application across various medical imaging tasks, supporting radiologists with

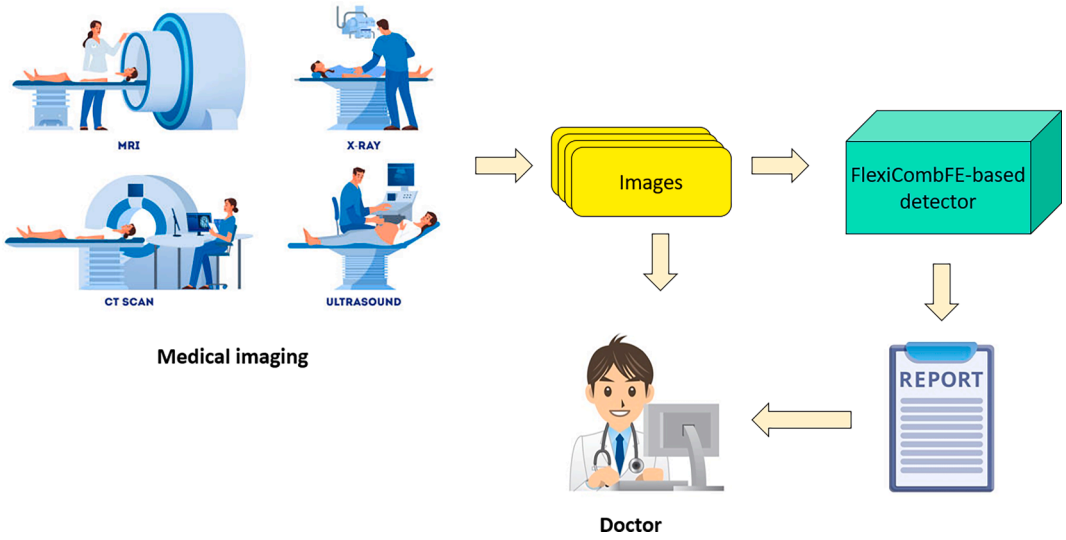

Fig. 14. Denotation of the FlexiCombFE-based medical imaging pipeline. Medical images such as MRI, X-ray, CT scan, and ultrasound are used to capture diagnostic images. These images are processed by the FlexiCombFE-based detector. The final report aids the doctor in interpreting the results and making informed clinical decisions.

accurate and clear diagnoses.

As illustrated in Fig. 14, the FlexiCombFE-based medical imaging pipeline processes various medical images (MRI, X-ray, CT scan, ultrasound) through its detection system, providing reports that aid doctors in making informed clinical decisions.

6. Conclusions

The introduced FlexiCombFE framework has showcased exceptional adaptability and precision in handling diverse image characteristics for brain tumor classification. Our analysis validated the effectiveness of combining LPQ and PHOG as feature extractors, which significantly contribute to generating the most accurate outcomes. The use of the NCA feature selector consistently supported the identification of the most relevant features, contributing to the framework's high classification accuracy. The choice of a 70 × 70 patch size proved optimal, capturing sufficient image detail while maintaining computational efficiency. The incorporation of IMV and greedy algorithms in the information fusion phase further improved classification accuracy. These strategies ensured that the final classification decision leveraged the strengths of individual classifiers. The FlexiCombFE framework's performance, characterized by its 99.35 % accuracy, exemplifies the potential of combining traditional feature engineering techniques with innovative classification and fusion methods. This success in brain tumor image classification sets a precedent for applying similar frameworks to other complex image classification tasks, offering a promising direction for future research.

Ethical approval.

Not applicable.

Data availability: This dataset is available in Kaggle and can be downloaded using https://www.kaggle.com/datasets/preetviradiya/brian-tumor-dataset URL.

CRediT authorship contribution statement

Ilknur Tuncer: Writing – original draft, Visualization, Validation, Software, Methodology, Data curation. **Abdul Hafeez Baig:** Conceptualization, Methodology, Validation, Formal analysis, Investigation, Resources, Data curation, Visualization, Writing – review & editing. **Prabal Datta Barua:** Software, Conceptualization, Methodology, Validation, Formal analysis, Investigation, Resources, Data curation, Visualization, Writing – review & editing. **Rena Hajiyeva:** Writing – review & editing, Resources. **Salvi Massimo:** Conceptualization, Methodology, Validation, Formal analysis, Investigation, Resources, Data curation, Visualization, Writing – review & editing. **Sengul Dogan:** Writing – original draft. **Turker Tuncer:** Writing – review & editing, Supervision. **U.R. Acharya:** Writing – review & editing, Supervision, Conceptualization.

Funding

The authors state that this work has not received any funding.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

None

Data availability

Data will be made available on request.

References

[1] A.V. Ravi Kiran, G.K. Kumari, P.T. Krishnamurthy, A.P. Johnson, M. Kenchegowda, R.A.M. Osmani, A.S. Abu Lila, A. Moin, H. Gangadharappa, S.M.D. Rizvi, An update on emergent nano-therapeutic strategies against pediatric brain tumors, Brain Sci. 14 (2024) 185.
[2] P. Kleihues, P.C. Burger, B.W. Scheithauer, The new WHO classification of brain tumours, Brain Pathol. 3 (1993) 255–268.
[3] T.T. Lah, M. Novak, B. Breznik, Brain malignancies: glioblastoma and brain metastases, Seminars in Cancer Biology. Elsevier (2020) 262–273.
[4] Y. Gidron, Brain Tumor, Encyclopedia of Behavioral Medicine. Springer (2020) 292–293.
[5] Rahman, M.S., Suresh, S., Waly, M.I., 2018. Risk factors for cancer: Genetic and environment. Bioactive Components, Diet and Medical Treatment in Cancer Prevention, 1-23.
[6] Q.T. Ostrom, M. Adel Fahmideh, D.J. Cote, I.S. Muskens, J.M. Schraw, M. E. Scheurer, M.L. Bondy, Risk factors for childhood and adult primary brain tumors, Neuro Oncol. 21 (2019) 1357–1375.
[7] A. Vienne-Jumeau, C. Tafani, D. Ricard, Environmental risk factors of primary brain tumors: a review, Rev. Neurol. 175 (2019) 664–678.
[8] WHO, 2020. Cancer, World Health Organization, https://www.who.int/news-room/fact-sheets/detail/cancer.
[9] J.R. McFaline-Figueroa, E.Q. Lee, Brain tumors, Am. J. Med. 131 (2018) 874–882.

[10] K.J. Zülch, Brain tumors: their biology and pathology, Springer-Verlag, 2013.
[11] L.M. DeAngelis, Brain tumors, N. Engl. J. Med. 344 (2001) 114–123.
[12] S. Lapointe, A. Perry, N.A. Butowski, Primary brain tumours in adults, Lancet 392 (2018) 432–446.
[13] S.E. Dilsizian, E.L. Siegel, Artificial intelligence in medicine and cardiac imaging: harnessing big data and advanced computing to provide personalized medical diagnosis and treatment, Curr. Cardiol. Rep. 16 (2014) 1–8.
[14] J.S. Duncan, N. Ayache, Medical image analysis: progress over two decades and the challenges ahead, IEEE Trans. Pattern Anal. Mach. Intell. 22 (2000) 85–106.
[15] H.P. Chan, L.M. Hadjiiski, R.K. Samala, Computer-aided diagnosis in the era of deep learning, Med. Phys. 47 (2020) e218–e227.
[16] A. Chattopadhyay, M. Maitra, MRI-based brain tumour image detection using CNN based deep learning method, Neurosci. Inf. 2 (2022) 100060.
[17] K.-R. Müller, M. Tangermann, G. Dornhege, M. Krauledat, G. Curio, B. Blankertz, Machine learning for real-time single-trial EEG-analysis: from brain–computer interfacing to mental state monitoring, J. Neurosci. Methods 167 (2008) 82–90.
[18] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). IEEE (2005) 886–893.
[19] A. Saïdani, A.K. Echi, Pyramid histogram of oriented gradient for machine-printed/handwritten and Arabic/Latin word discrimination, 2014 6th International Conference of Soft Computing and Pattern Recognition (SoCPaR). IEEE (2014) 267–272.
[20] V. Ojansivu, J. Heikkilä, Blur insensitive texture classification using local phase quantization, International Conference on Image and Signal Processing. Springer (2008) 236–243.
[21] T. Ojala, M. Pietikainen, T. Maenpaa, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, IEEE Trans. Pattern Anal. Mach. Intell. 24 (2002) 971–987.
[22] J. Goldberger, G.E. Hinton, S. Roweis, R.R. Salakhutdinov, Neighbourhood components analysis, Adv. Neural Inf. Proces. Syst. 17 (2004) 513–520.
[23] H. Liu, R. Setiono, Chi2: Feature selection and discretization of numeric attributes, in: Proceedings of 7th IEEE International Conference on Tools with Artificial Intelligence. IEEE, 1995, pp. 388–391.
[24] I. Kononenko, Estimating attributes: analysis and extensions of RELIEF, European Conference on Machine Learning. Springer (1994) 171–182.
[25] J. Maillo, S. Ramírez, I. Triguero, F. Herrera, kNN-IS: An Iterative Spark-based design of the k-Nearest Neighbors classifier for big data, Knowl.-Based Syst. 117 (2017) 3–15.
[26] V. Vapnik, The support vector method of function estimation, Nonlinear Modeling. Springer (1998) 55–85.
[27] T. Rahman, M.S. Islam, MRI brain tumor detection and classification using parallel deep convolutional neural networks, Meas.: Sens. 26 (2023) 100694.
[28] S. Patil, D. Kirange, Ensemble of deep learning models for brain tumor detection, Procedia Comput. Sci. 218 (2023) 2468–2479.
[29] N. Ullah, M. Hassan, J.A. Khan, M.S. Anwar, K. Aurangzeb, Enhancing explainability in brain tumor detection: a novel DeepEBTDNet model with LIME on MRI images, Int. J. Imaging Syst. Technol. 34 (2024) e23012.
[30] F.B. Demir, M. Baygin, I. Tuncer, P.D. Barua, S. Dogan, T. Tuncer, C.P. Ooi, E. J. Ciaccio, U.R. Acharya, MNPDenseNet: automated monkeypox detection using multiple nested patch division and pretrained DenseNet201, Multimed. Tools Appl. (2024) 1–23.
[31] T. Kivrak, J. Nayak, M.A. Gelen, P.D. Barua, M. Baygin, H.E. Pamukcu, S. Dogan, T. Tuncer, U.R. Acharya, EfDenseNet: Automated pulmonary hypertension detection model based on EfficientNetb0 and DenseNet201 using CT images, IEEE Access (2023).
[32] S. Tatli, G. Macin, I. Tasci, B. Tasci, P.D. Barua, M. Baygin, T. Tuncer, S. Dogan, E. J. Ciaccio, U.R. Acharya, Transfer-transfer model with MSNet: an automated accurate multiple sclerosis and myelitis detection system, Expert Syst. Appl. 236 (2024) 121314.
[33] I. Lauriola, A. Lavelli, F. Aiolli, An introduction to deep learning in natural language processing: models, techniques, and tools, Neurocomputing 470 (2022) 443–456.
[34] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., 2020. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv: 2010.11929.
[35] J. Maurício, I. Domingues, J. Bernardino, Comparing vision transformers and convolutional neural networks for image classification: a literature review, Appl. Sci. 13 (2023) 5521.
[36] Ranftl, R., Bochkovskiy, A., Koltun, V., 2021. Vision transformers for dense prediction, Proceedings of the IEEE/CVF international conference on computer vision, pp. 12179-12188.
[37] Wu, B., Xu, C., Dai, X., Wan, A., Zhang, P., Yan, Z., Tomizuka, M., Gonzalez, J., Keutzer, K., Vajda, P., 2020. Visual transformers: Token-based image representation and processing for computer vision. arXiv preprint arXiv: 2006.03677.
[38] P. Lv, W. Wu, Y. Zhong, F. Du, L. Zhang, SCViT: a spatial-channel feature preserving vision transformer for remote sensing image scene classification, IEEE Trans. Geosci. Remote Sens. 60 (2022) 1–12.
[39] A.M. Yildiz, P.D. Barua, S. Dogan, M. Baygin, T. Tuncer, C.P. Ooi, H. Fujita, U. R. Acharya, A novel tree pattern-based violence detection model using audio signals, Expert Syst. Appl. 224 (2023) 120031.
[40] Viradiya, P., 2021. Brian Tumor Dataset, https://www.kaggle.com/datasets/preetviradiya/brian-tumor-dataset.
[41] S. Solanki, U.P. Singh, S.S. Chouhan, S. Jain, Brain tumor detection and classification using intelligence techniques: an overview, IEEE Access (2023).
[42] S. Saeedi, S. Rezayi, H. Keshavarz, SR. Niakan Kalhori, MRI-based brain tumor detection using convolutional deep learning methods and chosen machine learning techniques, BMC Med. Inf. Decis. Making 23 (2023) 16.
[43] M. Aamir, Z. Rahman, W.A. Abro, U.A. Bhatti, Z.A. Dayo, M. Ishfaq, Brain tumor classification utilizing deep features derived from high-quality regions in MRI images, Biomed. Signal Process. Control 85 (2023) 104988.
[44] S. Hossain, A. Chakrabarty, T.R. Gadekallu, M. Alazab, M.J. Piran, Vision transformers, ensemble model, and transfer learning leveraging explainable AI for brain tumor detection and classification, IEEE J. Biomed. Health Inform. (2023).
[45] M.A. Mahjoubi, S. Hamida, O. Gannour, B. Cherradi, A. Abbassi, A. Raihani, Improved multiclass brain tumor detection using convolutional neural networks and magnetic resonance imaging, Int. J. Adv. Comput. Sci. Appl. 14 (2023) 406–414.
[46] M.I. Mahmud, M. Mamun, A. Abdelgawad, A deep analysis of brain tumor detection from mr images using deep learning networks, Algorithms 16 (2023) 176.
[47] S. Kumar, S. Choudhary, A. Jain, K. Singh, A. Ahmadian, M.Y. Bajuri, Brain tumor classification using deep neural network and transfer learning, Brain Topogr. 36 (2023) 305–318.
[48] A.B. Abdusalomov, M. Mukhiddinov, T.K. Whangbo, Brain tumor detection based on deep learning approaches and magnetic resonance imaging, Cancers 15 (2023) 4172.
[49] O. Özkaraca, O.İ. Bağrıaçık, H. Gürüler, F. Khan, J. Hussain, J. Khan, U.e. Laila, Multiple brain tumor classification with dense CNN architecture using brain MRI images, Life 13 (2023) 349.
[50] A.K. Sharma, A. Nandal, A. Dhaka, L. Zhou, A. Alhudhaif, F. Alenezi, K. Polat, Brain tumor classification using the modified ResNet50 model based on transfer learning, Biomed. Signal Process. Control 86 (2023) 105299.
[51] S. Asif, M. Zhao, F. Tang, Y. Zhu, An enhanced deep learning method for multi-class brain tumor classification using deep transfer learning, Multimed. Tools Appl. 82 (2023) 31709–31736.
[52] P. Kanchanamala, K. Revathi, M.B.J. Ananth, Optimization-enabled hybrid deep learning for brain tumor detection and classification from MRI, Biomed. Signal Process. Control 84 (2023) 104955.
[53] T. Muezzinoglu, N. Baygin, I. Tuncer, P.D. Barua, M. Baygin, S. Dogan, T. Tuncer, E.E. Palmer, K.H. Cheong, U.R. Acharya, PatchResNet: multiple patch division–based deep feature fusion framework for brain tumor classification using MRI images, J. Digit. Imaging 36 (2023) 973–987.
[54] E. Kaplan, W.Y. Chan, H.B. Altinsoy, M. Baygin, P.D. Barua, S. Chakraborty, S. Dogan, T. Tuncer, U.R. Acharya, PFP-HOG: pyramid and fixed-size patch-based HOG technique for automated brain abnormality classification with MRI, J. Digit. Imaging 36 (2023) 2441–2460.
[55] M. Woźniak, J. Siłka, M. Wieczorek, Deep neural network correlation learning mechanism for CT brain tumor detection, Neural Comput. & Applic. 35 (2023) 14611–14626.
[56] M.Z. Khaliki, M.S. Başarslan, Brain tumor detection from images and comparison with transfer learning methods and 3-layer CNN, Sci. Rep. 14 (2024) 2664.
[57] D. Bala, M.A. Islam, M.I. Hossain, M. Mynuddin, M.A. Hossain, M.S. Hossain, Automated brain tumor classification system using convolutional neural networks from mri images, in: 2022 International Conference on Engineering and Emerging Technologies (ICEET). IEEE, 2022, pp. 1–6.
[58] Jain, S., Jaidka, P., 2023. Brain Tumor Classification using Deep learning based Novel Hybrid Approach, 2023 2nd International Conference on Edge Computing and Applications (ICECAA). IEEE, pp. 872-876.
[59] W. Chmiel, J. Kwiecień, K. Motyka, Saliency map and deep learning in binary classification of brain tumours, Sensors 23 (2023) 4543.
[60] J.C.V. Gómez, A.P.Z. Incalla, J.C.C. Perca, D.I.M. Padilla, Diferentes configuraciones para MobileNet en la detección de tumores cerebrales: Different configurations for MobileNet in the detection of brain tumors, in: 2021 IEEE 1st International Conference on Advanced Learning Technologies on Education & Research (ICALTER). IEEE, 2021, pp. 1–4.
[61] O.S. Ads, M.M. Alfares, M.-A.-M. Salem, Multi-limb split learning for tumor classification on vertically distributed data, in: 2021 Tenth International Conference on Intelligent Computing and Information Systems (ICICIS). IEEE, 2021, pp. 88–92.
[62] Z. Chen, Study of transferability of imagenet-based pretrained model to brain tumor MRI dataset, in: 2022 International Conference on Machine Learning and Intelligent Systems Engineering (MLISE). IEEE, 2022, pp. 87–91.