Engineering and Technology Journal
Journal homepage: https://etj.uotechnology.edu.iq

# The effect of synthetic minority oversampling for enhancing brain tumor image classification

**Farah R. Fadhil*, Zainab N. Sultani**
Computer Science, College of Science, Al-Nahrain University, Baghdad, Iraq.
*Corresponding author Email: st.farahraad22@ced.nahrainuniv.edu.iq

## HIGHLIGHTS

- Two MRI brain tumor datasets were used, covering glioma, meningioma, pituitary, and healthy cases.
- Images were preprocessed with resizing, sharpening, CLAHE, and Otsu thresholding for clarity.
- GLCM and HOG features, along with their combination, were extracted to capture texture and area.
- SMOTE was applied to balance class distributions and enhance classifier performance.

**Keywords:**

Machine learning
Brain tumor MRI
Feature extraction
Data augmentation
Image classification.

## ABSTRACT

Brain tumors are among the most serious neurological diseases, posing significant diagnostic challenges due to their diverse nature and complexity imbalance in medical imaging datasets. To address these challenges, machine learning (ML) has demonstrated high potential in brain tumor classification; nevertheless, its overall performance can suffer when minority classes are underrepresented. This study examines the impact of the Synthetic Minority Over-sampling Technique (SMOTE) on MRI brain tumor classification using handcrafted features, including Gray Level Co-occurence Matrix (GLCM), Histogram of Oriented Gradients (HOG), and their combination (HOG + GLCM), with three classifiers, namely, Logistic Regression (LR), Support Vector Machines (SVM), and k-Nearest Neighbours (KNN). Experiments were carried out using two publicly available datasets. For Dataset 1, SVM with GLCM increased from 55.21 to 57.40% (full SMOTE), but LR with HOG + GLCM increased from 52.4 to 53.49%. For Dataset 2, fold-wise SMOTE increased LR with GLCM from 69.85 to 70.60% and SVM with HOG increased from 92.87 to 93.10% compared to complete SMOTE. The results show that the SMOTE's effect is dependent on feature type, classifier, and augmentation strategy, with fold-wise typically boosting generalization while avoiding information leaking. These findings validate SMOTE as a viable method for improving the type overall performance in imbalanced medical imaging tasks, particularly for weaker texture-based descriptors.

## 1. Introduction

Image classification is the process of classifying and labelling images utilizing common features found in images that belong to different classes [1]. In this regard, classifying medical images plays an essential role in biotechnology because it improves diagnostic and clinical decision-making [2]. However, the primary obstacles in image classification, despite its increasing significance, are the sheer number of images, the intricacy of the data, and the scarcity of labeled data. In particular, it is difficult to develop dependable and proper classification systems because of these issues [3], especially when it comes to medical image classification, in which there is frequently not enough data available to train reliable classifiers, which makes it more difficult to increase the classification accuracy [2].

In this context, the widespread use of data augmentation is mostly attributable to its positive effects on generalization, or the ML models' capacity to correctly predict the outcomes concerning data that was not encountered during training [4]. Other advantages of DA have also been shown, including enhanced resilience to transformations and support for model calibration [3], time, and resources. Moreover, it can increase the training dataset without requiring the acquisition and labelling of new natural data [4].

To this end, traditional data augmentation methods for classification in medical imaging include techniques such as flipping, adding noise, and applying geometric transformations [6]. Additional methods include brightness, adjusting saturation, and contrast, which all offer fresh insights into the same information. These adjustments handle issues with backdrop scaling, lighting variations, occlusion, and perspective. Additionally, data augmentation enhances the model's diversity and acts as a regularizer, thereby improving its capacity to generalize to new data and reducing overfitting [5].



http://doi.org/10.30684/etj.2025.163902.2001
Received 10 August 2025; Received in revised form 25 August 2025; Accepted 31 August 2025; Available online 28 September 2025
2412-0758/University of Technology-Iraq, Baghdad, Iraq

In addition to overfitting, imbalanced data is a common problem for machine learning models. Techniques such as the Synthetic Minority Over-sampling Technique (SMOTE) have been proposed to address this issue. Here, data augmentation works especially well for managing unbalanced datasets [7,4]. More specifically, data augmentation can artificially increase the representation of minority classes in imbalanced datasets, where the number of examples in one class (majority) greatly exceeds that of other classes, which improves the model's capacity to learn from underrepresented data [4]. By identifying the closest neighbors of minority class instances and drawing synthetic samples along their borders, the SMOTE produces synthetic data [7].

By analyzing how DA affects ML models with imbalanced data, it becomes possible to isolate and observe the improvements directly. In such cases, DA serves not only to expand the dataset but also to ensure that minority classes are adequately represented during the training process [4]. This step yields enhanced classification performance, particularly for practical applications where the costs of misclassifying minority examples are substantial [7].

The importance of data augmentation in boosting global scientific image classification performance across diverse modalities and architectures was investigated by Rama and Nalini [3]. In this study, convolutional neural networks (CNNs) were used to identify lung X-ray images, demonstrating a step forward in classification accuracy. At the same time, various augmentation approaches based on shear differences were implemented, thereby reaching a validation accuracy of up to 93%. Similarly, TensorMixup, a feature-based augmentation method, was suggested by Wang et al. [8], and they used the 3D U-Net architecture to improve brain tumor segmentation. The dual-stage training procedure, which used both conventional and TensorMixup-augmented data, produced remarkable dice scores on the BraTS2019 dataset, specifically 91.32% for the entire tumor segmentation. In another work, Wang et al. [9], focused on few-shot image classification and the IFR (Information Fusion Rectification) approach, which uses cosine similarity to match question characteristics with relevant base magnificence prototypes. This strategy dramatically improved the accuracy on benchmark datasets, such as miniImageNet and CUB, utilizing classifiers including SVM, logistic regression, and MLP. This study emphasized the importance of feature refining and fusion in improving learning from limited data.

In the medical field, various studies have investigated the way augmentation and feature engineering improve classification accuracy. In particular, various augmentation methods for imbalanced datasets were tested using logistic regression, SVM, and CNN classifiers by Dablain and Chawla [4]. Their findings showed that advanced methods, such as ReMix, DeepSMOTE, and EOS, can significantly improve performance in general, with EOS achieving an accuracy of 0.796. Moreover, the effects of feature-level augmentation methods on breast cancer classification were investigated by Hasan et al., [10]. A comparative study of deep GoogleNet features along with Haralick capabilities revealed that the Mixup-based total augmentations combined with Haralick functions had the greatest AUC of 0.929.

Furthermore, the importance of transfer learning (TL) over multi-label medical image classification was emphasized by Alam et al. [11], using pre-skilled architectures such as ResNet50 and DenseNet201 on retinal and brain tumor datasets. They achieved significant improvements in accuracy, sensitivity, and specificity by combining conventional augmentation with the SMOTE and traditional machine learning methods. Based on the above discussion, Table1 provides a comparative assessment of recent studies that investigated various information augmentation strategies performed in unique medical and non-medical image classification scenarios. More precisely, this table emphasizes the augmentation techniques employed, the topic of each study, and the suggested performance indicators.

**Table 1:** The comparative analysis between multiple studies that use data augmentation

| Focus | Data augmentation method | Results | Ref. |
|---|---|---|---|
| Classification of Lung X-ray images | Shear-based augmentation techniques. | Validation accuracy up to 93%. | [3] |
| Brain tumor segmentation | TensorMixup combined with conventional augmentation in dual-stage training. | Dice score of 91.32% for the entire tumor segmentation (BraTS2019). | [8] |
| Few-shot image classification | Feature refinement with the IFR approach combined with cosine similarity and limited augmentation. | Significant accuracy improvement on miniImageNet and CUB. | [9] |
| Handling imbalanced medical datasets | ReMix, DeepSMOTE, and EOS augmentation methods. | EOS achieved an accuracy of 0.796. | [4] |
| Breast cancer classification | Mixup-based augmentation combined with Haralick features and GoogleNet deep features. | The highest AUC of 0.929. | [10] |
| Multi-label medical image classification | Traditional augmentation + SMOTE integrated with transfer learning (ResNet50, DenseNet201). | Significant improvement in accuracy, sensitivity, and specificity. | [11] |

However, while prior research has demonstrated the efficacy of several data augmentation strategies for medical image classification and segmentation, significant hurdles remain. More specifically, most studies focused on spatial or pixel-level alterations (e.g., flipping, rotation, and noise addition) that do not adequately address the problem of class imbalance, which is quite common in medical imaging datasets, particularly MRI scans.

In terms of brain tumor classification, existing augmentation attempts have generally focused on geometric or volume-based alterations. In contrast, limited emphasis has been paid to oversampling approaches that immediately correct the imbalance

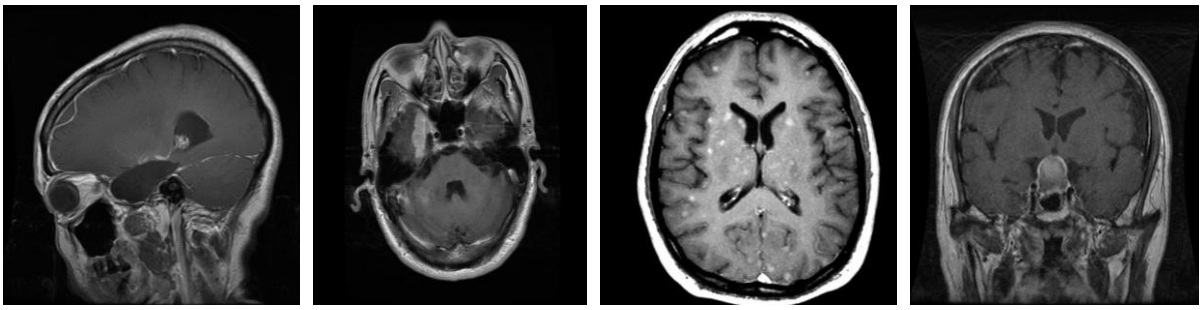
between tumor categories. Despite the recognized benefit of resampling approaches in increasing the classifier stability and generalization, their application in MRI-based tumor classification remains unexplored.

To overcome these obstacles, this work investigates the application of the Synthetic Minority Oversampling Technique (SMOTE) for improving the classification of brain tumor MRI images. Specifically, SMOTE aims to stabilize the dataset distribution, reduce bias in model training, and improve typical classification performance by creating synthetic samples of underrepresented tumor classifications. Accordingly, this technique provides a feature-level augmentation framework specifically tailored for brain MRI classification.

## 2. Dataset

The two publicly available brain MRI datasets from Kaggle were used in this study. Both of these datasets contain images from four different classes, including glioma, meningioma, pituitary, and no tumor.

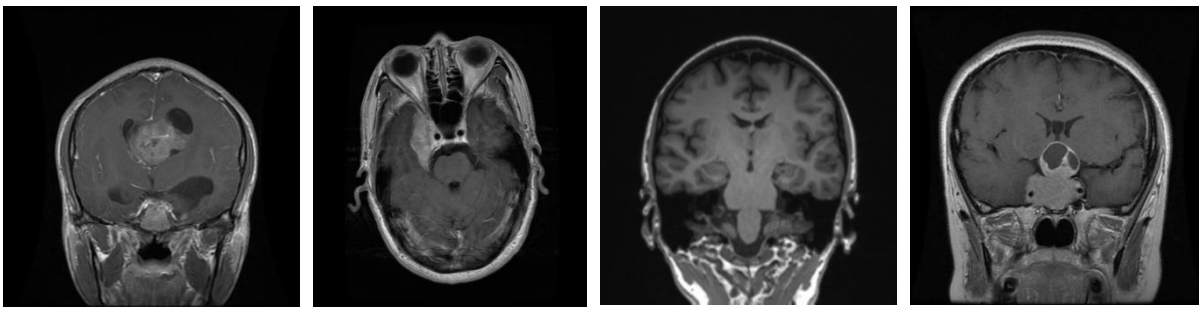
The first dataset provides 3,264 MRI images. The images are separated into two sets: train and test (394 images each). The training distribution is imbalanced, with 822 meningiomas, 395 no tumors, 827 pituitary, and 826 gliomas. This disparity makes the dataset particularly difficult. Figure 1 illustrates samples of data for each class, where (a) represents Glioma, (b) represents Meningioma, (c) represents Healthy, and (d) represents Pituitary.



**(a)** **(b)** **(c)** **(d)**

**Figure 1:** Images in four classes: (a) Glioma, (b) Meningioma, ( c ) Healthy, (d) Pituitary

The second dataset is similar to the first one, with four categories: glioma (1321), meningioma (1339), pituitary (1457), and no tumor (1595). The total number of images in the test data is 1311. When compared to the primary dataset, it contains a greater number of higher-quality photos. It provides a more equal class distribution, which allowed us to examine the suggested method's generalizability across unbiased record assets. Figure 2 illustrates samples for each class, where (a) represents Glioma, (b) represents Meningioma, (c) represents Healthy, and (d) represents Pituitary.



**(a)** **(b)** **(c)** **(d)**

**Figure 2:** Images in four classes: (a) Glioma, (b) Meningioma, ( c ) Healthy, (d) Pituitary

## 3. Methodology

This section describes the overall block diagram for the process of identifying MRI brain tumors using typical machine learning techniques, as shown in Figure 3. More specifically, image preprocessing is the first step, followed by segmentation using Otsu's thresholding. Grey Level Cooccurrence Matrix (GLCM), Histogram Oriented Gradient (HOG), or a combination of them (HOG + GLCM) is employed to extract features. In addition, standardization is performed before classification, and the SMOTE is used to train the features to alleviate class imbalance. Classification is performed using logistic regression (LR), support vector machines (SVM), and k-nearest neighbors (KNN). In practice, three experimental configurations were considered, including individual features, combined features, and SMOTE-augmented features. To achieve optimal and balanced performance, stratified k-fold cross-validation with GridSearchCV was employed for model training and hyperparameter optimization. Additionally, the model's performance was evaluated using standard metrics, including accuracy, precision, recall, and F1-score.

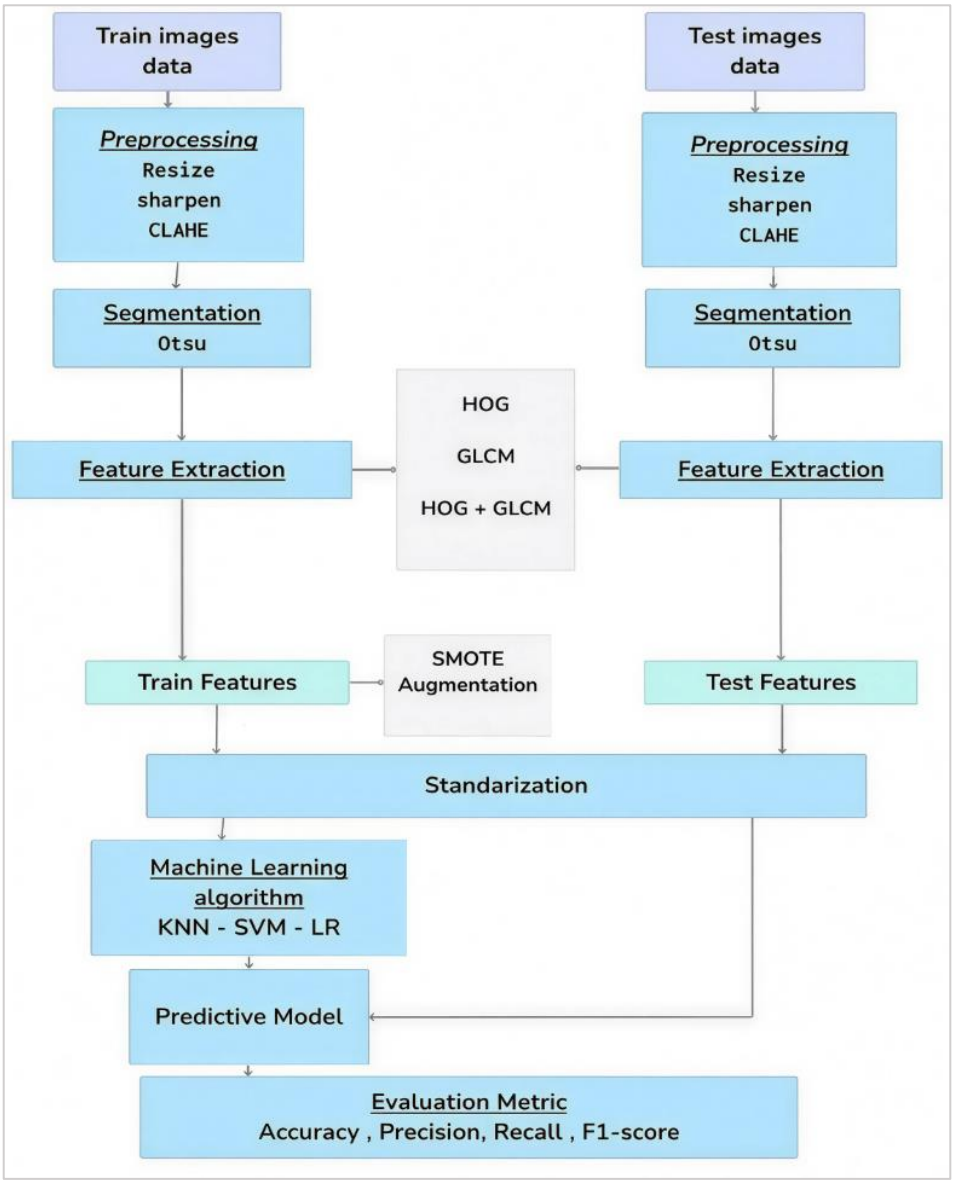



**Figure 3:** Block diagram of MRI brain tumor classification

## 3.1 Image preprocessing

Noise-induced transmission image processing degrades brain MRIs during imaging. The preprocessing procedure improves image quality and contrast by reducing noise and fluctuation from the brain MRI. This section presents a quick summary of the preprocessing methods utilized in this study [12].

### 3.1.1 Resize

The collection of MRI images varies in size. Therefore, to maintain consistency, all photos were downsized to 128 × 128 pixels using bilinear interpolation, which preserves key visual elements with fewer artefacts, is more computationally efficient, and generates smoother transitions compared to the nearest-neighbor [13].

### 3.1.2 Sharpen with gaussian

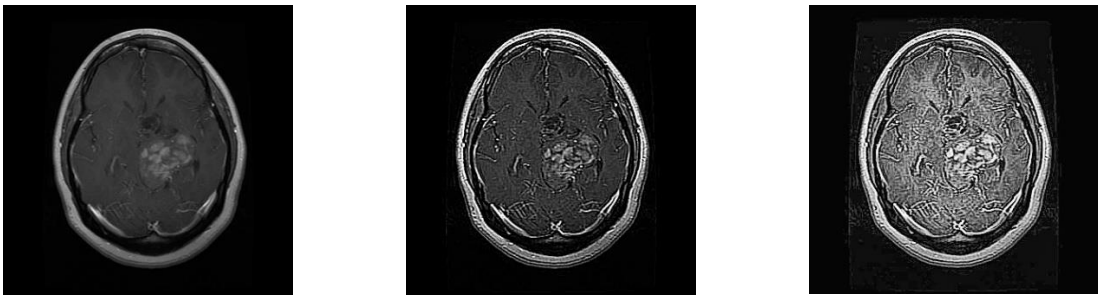
Following resizing, image sharpening was used to enhance fine details, minimize noise, and improve the quality of features extracted for classification tasks [14]. The following formula was used to blend the original image with a blurred version created by applying a Gaussian filter :

$$\text{Sharpened Image} = 7 \times \text{Original Image} - 6 \times \text{Blurred Image} \quad (1)$$

Equation 1 uses $\beta = -6$ to reduce noise in the blurred image and $\alpha = 7$ to improve the details in the original image. Specifically, a Gaussian filter was employed with a kernel size of (5,5) and a sigma value of (3) [15].

### 3.1.3 Contrast-limited adaptive histogram equalization

Throughout the sharpening process, the Contrast-Limited Adaptive Histogram Equalisation (CLAHE) was used to enhance feature visibility while lowering noise amplification. By spatially separating the image into tiny parts and equalizing each one separately [16], the CLAHE boosts contrast in low-intensity areas in a different way than the standard histogram equalization. The chosen parameters are the tile grid size (10×10), which determines the number of image subdivisions for localized contrast corrections, and the clip limit (4), which controls the level of contrast enhancement [15]. Figure 4 depicts the several stages of image preparation that were used in this investigation, where (a) represent original image , (b) represent sharpened image , (c) represent CLAHE image



**(a) Original** **(b) Sharpen** **(c) CLAHE**

**Figure 4:** The impact of sharpening with the Gaussian and the CLAHE sequentially

## 3.2 Image segmentation

This particular type of segmentation relies on intensity. Thresholding, often known as image binarization, is an important image processing technique. It removes the object from the background. When compared to grey-level images that typically contain an enormous number of grey levels (up to 256 levels), segmented images obtained through thresholding have the advantages of smaller storage space, faster processing speed, and ease of manipulation. This step produces a segmented image with a black background and a bright tumor area [17].

In this context, Otsu's thresholding is a common approach for automatically determining the differences in intensity between two sets of pixels in an image. It accomplishes this process by determining the best way to divide the two groups. In the detection of brain tumors, Otsu's approach can be used to identify tumorous regions from healthy brain tissue using MRI scans [18]. The Otsu thresholding technique involves calculating a spread measure for both sides of the pixel concentration limit, which refers to the pixels in the foreground or background. Otsu's method usually selects the threshold by minimizing the in-class variance of the two pixel groups that the operator separates [19]. Figure 5 illustrates the effect of the Otsu threshold on the image.

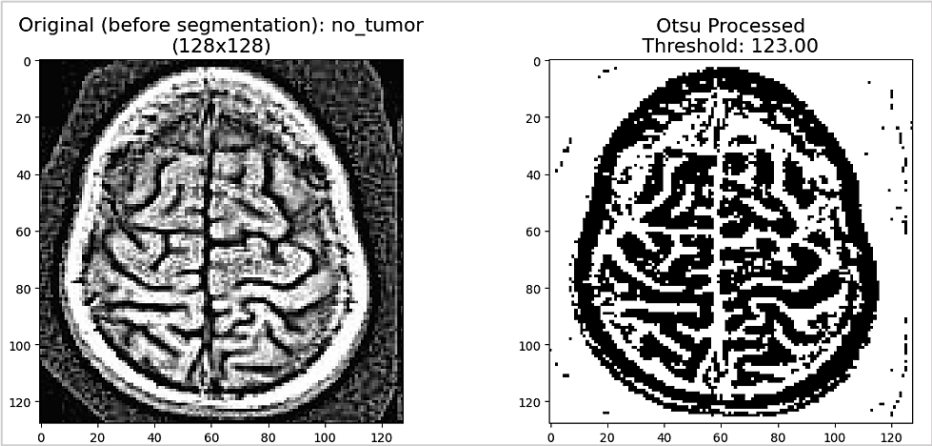



**Figure 5:** The effect of the Otsu thresholding

## 3.3 Feature extraction

Converting the image to its useful properties is called feature extraction. The extracted features from the preprocessed and segmented images are used as input to the ML algorithms. In this regard, obtaining a useful number of features from brain MRI images is a very challenging task [20]. In this study, three methods were applied separately, including the Histogram Oriented Gradient (HOG), the Grey Level Co-occurrence Matrix (GLCM), and a combination of the HOG and the GLCM. These methods are described in detail as follows:

### 3.3.1 Histogram oriented gradient (HOG)

The Histogram of Oriented Gradients (HOG) measures the change in gradient orientations and is used to identify an image's structure and local shape [21]. In particular, the images used in this study are broken into 24 × 24 pixel cells and arranged into 5

× 5 pixel blocks. The gradient directions are quantified, yielding six bins. Histograms are generated and normalized inside each block to improve robustness towards variations in illumination and contrast. The final descriptor is obtained by concatenating each of the local histograms. The summarized steps of applying this feature extraction type are as follows:

1. The target image is segmented into square cells with defined dimensions.
2. The horizontal ($g_x$) and vertical ($g_y$) Gradients for each cell are generated using derivative masks [-1, 0, 1] and [-1, 0, 1]$^T$, respectively. Using ($g_x$) and ($g_y$). The magnitude and the direction of the gradient can be calculated as shown in Equations 1 and 2.

$$\text{gradient magnitude} = \sqrt{g_x^2 + g_y^2} \tag{1}$$

$$\text{gradient direction} = \arctan \frac{g_y}{g_x} \tag{2}$$

A histogram vector is constructed for each image cell based on magnitude and direction matrices. The histogram bins are determined based on gradient direction, and the values within them are a cumulative weighted vote based on the gradient magnitude. Figure 3 illustrates the process of filling the histogram bins. A set number of cell histograms is combined into a block whose values are normalised.

The method is repeated for all of the blocks that slide to cover the entire image, yielding a single big vector containing the retrieved features [22].

### 3.3.2 Grey level co-occurrence matrix (GLCM)

A Grey Level Co-occurrence Matrix (GLCM) is a statistical technique for texture analysis that considers the spatial correlations between pixels. The GLCM capabilities characterize the texture of an image [19]. Particularly, the GLCM expresses the frequency with which pairs of pixels with specified grey-level intensities appear, oriented and spaced apart from one another. This matrix is used to obtain second-order records [23]. The algorithm used in GLCM is illustrated in Algorithm 1. To compute the joint in this study, the GLCM is calculated using four distances (1, 2, 3, and 4 pixels) and four directions (0°, 45°, 90°, and 150°). Each matrix produced a set of statistical skills, including evaluation, dissimilarity, homogeneity, energy, correlation, maximum possibility, inertia, cluster colour, cluster prominence, and inverse difference. These functions are described using the following equations [24, 25].

1. Contrast: Equation 3 is used to calculate the contrast, which quantifies the intensity difference between the neighboring pixels in an image:

$$Contrast = \sum_{i,j=0}^{G-1} P_{ij}(i-j)^2 \tag{3}$$

The grey-level co-occurrence matrix (GLCM) shows the probability of transitioning between grey levels i and j as P(i,j).

2. Dissimilarity: Equation 4 is used to determine the dissimilarity feature:

$$Dissimilarity = \sum_{i,j=0}^{G-1} |i-j| P(i,j) \tag{4}$$

3. Homogeneity: Equation 5 is used to calculate the homogeneity, which shows how closely the distribution of the grey-level values is distributed:

$$Homogeneity = \sum_{i,j=0}^{G-1} \frac{1}{1+(i-j)^2} P(i,j) \tag{5}$$

4. Correlation: Equation 6 is used to calculate the correlation, which measures the linear relationship between the grey-level values:

$$Correlation = \sum_{i,j=0}^{G-1} \frac{\{(i \times j) \times P(i \times j)\} - (\mu_x - \mu_y)}{\sigma_x \times \sigma_y} \tag{6}$$

where $\mu_x$ and $\mu_y$ represent the mean grey-level values, and σx and σy denote the standard deviation.

5. The maximum probability: Equation 7 determines the maximum probability feature:

$$Maximum\ Probability = \binom{max}{i,j} P(i,j) \tag{7}$$

6. Entropy: Equation 8 is used to calculate the entropy, which represents the randomness in the grey-level distribution within the image:

$$Entropy = -\sum_{i,j=0}^{G-1} P(i,j) \times \log(P(i,j)) \tag{8}$$

7. Inertia: The variance within the GLCM is measured by the inertia, which is calculated using Equation 9:

$$Interia = \sum_{i=0}^{G-1}\sum_{j=0}^{G-1}(i-j)^2 P(i,j) \quad (9)$$

8. Clusters' prominence: Equation 10 is used to calculate this feature:

$$Cluster\ Prominance = \sum_{i,j=0}^{G-1}(i+j-\mu_x-\mu_y)^4 \times P(i,j) \quad (10)$$

9. Cluster shade: The cluster shade, which is determined using Equation 11, indicates how skewed the image's clusters are:

$$Cluster\ Shade = \sum_{i,j=0}^{G-1}(i+j-\mu_x-\mu_y)^3 \times P(i,j) \quad (11)$$

10. IDF, or Inverse Difference: Equation 12 shows how this feature is calculated:

$$Inverse\ Difference = \sum_{i,j=0}^{G-1}\frac{P(i,j)}{1+|i-j|} \quad (12)$$

**Algorithm .1. Grey Level Co-occurrence Matrix algorithm**

Input: Preprocessed color image of size 128 × 128
Output: GLCM feature vector

Convert the image to grayscale.
Convert the grayscale image to 8-bit unsigned integer format.
For each distance d∈{1,2,3,4}:
  For each angle θ∈{0°,45°,90°,135°}:
    a. Compute the co-occurrence matrix GLCM(d,θ) using normalized and symmetric settings.
    b. Extract the following features:
      – Contrast: Measures the intensity contrast between a pixel and its neighbor
      – Dissimilarity: Quantifies the difference between grey levels
      – Homogeneity: Measures the closeness of the distribution to the diagonal
      – Energy: Reflects textural uniformity (also known as Angular Second Moment)
      – Correlation: Assesses the linear dependency of grey levels
      – Max Probability: The highest probability in the GLCM matrix
      – Inertia: Similar to the contrast, but it is more sensitive to higher differences
      – Cluster Shade: Measures the skewness of the matrix
      – Cluster Prominence: Captures the sharpness or peakedness of the clusters
      – Inverse Difference: Emphasizes homogeneity in the texture
    c. Append all extracted features to the feature vector.
Return the concatenated feature vector.

### 3.3.3 The combination of HOG and GLCM (HOG + GLCM)

The combination of HOG and GLCM is considered [26]. The HOG feature captures the edge orientation patterns, and the GLCM features capture the spatial pixel intensity relationships. Next, the final feature vector for each image is produced by combining the original GLCM and HOG features. The performance of the image classification model could be improved by using this augmented feature set, which could offer a more comprehensive and nuanced image representation [27].

## 3.4 Synthetic minority oversampling technique (SMOTE)

The SMOTE algorithm is limited to minority class oversampling. This approach creates new samples by first calculating the k-nearest neighbours of the minority class samples, from which M samples (where M < k) are randomly selected to perform linear interpolation. Many k-nearest neighbours-based oversampling strategies neglect the majority class when selecting the nearest neighbours and creating fresh samples [28]. When working with uneven data, SMOTE helps create a more balanced dataset, which enhances the performance and robustness of machine learning models [29].

### 3.4.1 Implementation of SMOTE

**Step 1**: The SMOTE chooses the nearest neighbours for each minority sample $x_i (i=1,2,....,n)$ using a specified neighbour count number (in this case, k=4). The selection algorithm is based on the Euclidean distance between samples in the feature space.

**Step 2**: The over-sampling technique generates synthetic samples by selecting mm nearest neighbours at random from each minority sample's k-nearest neighbour set. The notation for each picked neighbour is $x_{i,j}$ $(j=1,2,....,m)$. Next, Equation 13 is applied for interpolation:

$$P_{i,j} = x_i + \text{rand}(0,1) \times (x_{i,j} - x_i), \quad (13)$$

An artificially generated minority sample $P_{i,j}$ is obtained, where rand (0,1) is a uniformly distributed random number in the range [0,1]. The technique is repeated until the dataset achieves the desired class balance. Finally, the modified distribution of class labels is evaluated before and after SMOTE to ensure balanced class representation in the training data [30].

### 3.4.2 Experimental application

For each feature type (HOG, GLCM, and HOG+GLCM), the SMOTE and other oversampling approaches (Borderline-SMOTE, SVM-SMOTE, and ADASYN) were employed to assess the influence of oversampling on classification performance. In this work, two different application configurations were investigated:

1. Total training augmentation (SMOTE): Before training the model, the complete training set is oversampled.
2. Fold-wise augmentation (SMOTE fold-wise): in cross-validation, oversampling is only performed in the training region of each fold, ensuring that the validation data remains unchanged.

To discover which setup performed best for each feature type, each approach was tested with different classifiers and parameter values.

## 3.5 Standardization

Standardization is applied to the retrieved features before classification to ensure uniformity across features. To guarantee that each feature contributes equally to the learning process, standardization translates feature values to a mean of 0 and a standard deviation of 1. This step is especially useful for models like Support Vector Machines (SVM) as well as k-Nearest Neighbours (KNN), which rely on distance-based computations [18].

## 3.6 Machine learning algorithms

This study employs three common supervised machine learning methods to accomplish multi-class classification tasks using extracted picture attributes [31]. These models were chosen because they are widely used, easily interpretable, and effective in medical image classification. In addition, a rigorous hyperparameter tuning method was performed for each algorithm using Grid Search and Stratified K-Fold cross-validation (CV=5), with the primary goal being the weighted F1-score.

### 3.6.1 K-Nearest neighbors (KNN)

The KNN is a non-parametric, instance-based learning algorithm that classifies new data points based upon the majority label of their k-nearest neighbours within the training set. The parameters chosen have a considerable impact on its performance [32]. More precisely, this paper compared several choices for the number of neighbours ($k \in \{3, 5, 7, 10, 15\}$), distance metrics (Euclidean, Manhattan, and Chebyshev), and weighting methods (uniform versus distance-based). Moreover, grid-based validation was performed to determine the best configuration. Table 2 illustrates the parameters chosen for each feature type for the first dataset, and Table 3 illustrates the parameters chosen for each feature type for the second dataset.

**Table 2:** KNN parameters for Dataset 1

| Feature type | Metric | Neighbors | Weights |
|---|---|---|---|
| GLCM | chebyshev | 5 | distance |
| GLCM SMOTE | chebyshev | 7 | distance |
| GLCM SMOTE fold-wise | chebyshev | 5 | distance |
| HOG | manhattan | 3 | distance |
| HOG SMOTE | manhattan | 3 | distance |
| HOG SMOTE fold-wise | manhattan | 3 | distance |
| HOG + GLCM | euclidean | 3 | distance |
| HOG + GLCM SMOTE | euclidean | 3 | distance |
| HOG + GLCM SMOTE fold-wise | euclidean | 3 | distance |

**Table 3:** KNN parameters for Dataset 2

| Feature type | Metric | Neighbors | Weights |
|---|---|---|---|
| GLCM | euclidean | 3 | distance |
| GLCM SMOTE | euclidean | 3 | distance |
| GLCM SMOTE fold-wise | euclidean | 3 | distance |
| HOG | manhattan | 3 | distance |
| HOG SMOTE | manhattan | 3 | distance |
| HOG SMOTE fold-wise | manhattan | 3 | distance |
| HOG + GLCM | euclidean | 3 | distance |
| HOG + GLCM SMOTE | euclidean | 3 | distance |
| HOG + GLCM SMOTE fold-wise | euclidean | 3 | distance |

### 3.6.2 Support vector machine (SVM)

SVM is a powerful classification model that divides data points into classes in a high-dimensional space, utilizing the ideal hyperplane. It performs exceptionally well for both linear and nonlinear classification problems [33]. In this paper, we considered four kernel functions: linear, polynomial, sigmoid, and radial basis function (RBF). The polynomial degree, the kernel coefficient, and the regularization parameter have been adjusted within predefined limits for non-linear kernels—the combination with the highest validation F1-score was discovered through a grid search. Table 4 illustrates the parameters chosen for each feature type for the first dataset, and Table 5 illustrates the parameters chosen for each feature type for the second dataset.

**Table 4:** SVM parameters for Dataset 1

| Feature type | Kernel | C | Gamma | Degree |
|---|---|---|---|---|
| GLCM | rbf | 100 | 0.01 | 1 |
| GLCM SMOTE | rbf | 100 | 0.01 | 1 |
| GLCM SMOTE fold-wise | sigmoid | 100 | 0.1 | 1 |
| HOG | rbf | 10 | 0.01 | 1 |
| HOG SMOTE | rbf | 10 | 0.01 | 1 |
| HOG SMOTE fold-wise | sigmoid | 100 | 0.1 | 1 |
| HOG + GLCM | rbf | 10 | 0.01 | 1 |
| HOG + GLCM SMOTE | rbf | 10 | 0.01 | 1 |
| HOG + GLCM SMOTE fold-wise | sigmoid | 100 | 0.1 | 1 |

**Table 5:** SVM parameters for Dataset 2

| Feature type | Kernel | C | Gamma | Degree |
|---|---|---|---|---|
| GLCM | rbf | 10 | 0.1 | 1 |
| GLCM SMOTE | rbf | 100 | 0.1 | 1 |
| GLCM SMOTE fold-wise | sigmoid | 100 | 0.1 | 1 |
| HOG | rbf | 10 | 0.01 | 1 |
| HOG SMOTE | rbf | 10 | 0.01 | 1 |
| HOG SMOTE fold-wise | sigmoid | 100 | 0.1 | 1 |
| HOG + GLCM | rbf | 10 | 0.01 | 1 |
| HOG + GLCM SMOTE | rbf | 10 | 0.01 | 1 |
| HOG + GLCM SMOTE fold-wise | sigmoid | 100 | 0.1 | 1 |

### 3.6.3 Logistic regression (LR)

Logistic regression is a popular linear model for binary and multiclass classification applications. It uses the logistic function to determine the chance of class membership [34]. In particular, penalties (L1, L2), regularisation strengths ($C$), maximum iteration limits, and solvers (lbfgs, Newton-cg, and Saga) were tested. The final setup was chosen based on the validation results and a cross-validated F1 score. Table 6 illustrates the parameters chosen for each feature type for the first dataset, and Table 7 illustrates the parameters chosen for each feature type for the second dataset.

**Table 6:** LR parameters for Dataset 1

| Feature type | C | Penalty | Solver | Max iter |
|---|---|---|---|---|
| GLCM | 10 | l2 | lbfgs | 1000 |
| GLCM SMOTE | 10 | l2 | lbfgs | 5000 |
| GLCM SMOTE fold-wise | 10 | l2 | saga | 5000 |
| HOG | 0.1 | l2 | saga | 1000 |
| HOG SMOTE | 0.1 | l2 | newton-cg | 1000 |
| HOG SMOTE fold-wise | 10 | l2 | saga | 5000 |
| HOG + GLCM | 1 | l1 | saga | 5000 |
| HOG + GLCM SMOTE | 1 | l1 | saga | 5000 |
| HOG + GLCM SMOTE fold-wise | 10 | l2 | saga | 5000 |

**Table 7:** LR parameters for Dataset 2

| Feature type | C | Penalty | Solver | Max iter |
|---|---|---|---|---|
| GLCM | 10 | l2 | lbfgs | 1000 |
| GLCM SMOTE | 10 | l2 | lbfgs | 5000 |
| GLCM SMOTE fold-wise | 10 | l2 | saga | 5000 |
| HOG | 0.1 | l2 | newton-cg | 1000 |
| HOG SMOTE | 0.1 | l2 | newton-cg | 1000 |
| HOG SMOTE fold-wise | 10 | l2 | saga | 5000 |
| HOG + GLCM | 10 | l2 | lbfgs | 1000 |
| HOG + GLCM SMOTE | 10 | l2 | lbfgs | 1000 |
| HOG + GLCM SMOTE fold-wise | 10 | l2 | saga | 5000 |

### 3.7 Evaluation metrics

Due to the imbalance in the dataset, accuracy alone is insufficient to assess model performance. Therefore, precision, recall, and F1-score metrics were also utilized to evaluate the model performance [35]. The formulas for measuring these metrics are as follows: Accuracy is given by Equation 14, Precision is defined in Equation 15, Recall is expressed in Equation 16, and the F1-Score is calculated using Equation 17.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{14}$$

$$\text{Precision} = \frac{TP}{TP+FP} \tag{15}$$

$$\text{Recall} = \frac{TP}{TP+FN} \tag{16}$$

$$\text{F1-Score} = \frac{2 \times precision \times recall}{precision+recall} \tag{17}$$

where the number of accurate positive predictions is indicated by TP (True Positive), the number of accurate negative predictions is indicated by TN (True Negative), and the number of inaccurate positive predictions is indicated by FP (False Positive). At the same time, FN is the (False Negative).

## 4. Results and discussion

This section provides an in-depth evaluation of the proposed technique, which explores the influence of SMOTE-based total augmentation on brain tumor photo classification using supervised learning models. In particular, three classifiers were used, including logistic regression (LR), support vector machine (SVM), and K-nearest neighbours (KNN). The studies were carried out using two curated brain MRI datasets to assess the type performance and generalizability of the models trained under different feature and augmentation settings.

In every dataset, the models were trained and evaluated utilizing three main feature configurations, as follows:

1. Original features (GLCM and HOG).
2. The SMOTE augmentation (SMOTE), which is the process of oversampling the entire training set using SMOTE before training.
3. Fold-wise SMOTE augmentation (SMOTE fold-wise), in which oversampling is performed most effectively within the training partition of each fold throughout cross-validation, leaving validation information undisturbed.

The primary goal is to determine how classes are balanced via the SMOTE that affects the classifier's overall performance, particularly test-time generalization, which is quantified by accuracy, precision, recall, and F1-score. Tables 7, 8, and 9 illustrate the classifiers' results of the first dataset, and Tables 10, 11, and 12 illustrate the classifiers' results of the second dataset.

**Table 8:** Summary results of the KNN classifier for Dataset 1

| | KNN | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Feature type** | **Train accuracy** | **Train precision** | **Train recall** | **Train F1** | **Val accuracy** | **Val precision** | **Val recall** | **Val F1** | **Test accuracy** | **Test precision** | **Test recall** | **Test F1** |
| GLCM | 100 | 100 | 100 | 100 | 75.50523 | 75.72496 | 75.50523 | 75.20321 | 78.93401 | 80.92714 | 78.93401 | 75.47375 |
| GLCM SMOTE | 100 | 100 | 100 | 100 | 79.29293 | 79.3219 | 79.29293 | 79.13085 | 75.88832 | 78.27566 | 75.88832 | 72.69135 |
| GLCM SMOTE fold-wise | 100 | 100 | 100 | 100 | 75.47038 | 75.42151 | 75.47038 | 75.23809 | 74.11168 | 76.52724 | 74.11168 | 70.72242 |
| HOG | 100 | 100 | 100 | 100 | 85.08711 | 85.09348 | 85.08711 | 84.74938 | 77.15736 | 80.79629 | 77.15736 | 72.56596 |
| HOG SMOTE | 100 | 100 | 100 | 100 | 87.84699 | 87.87516 | 87.84699 | 87.52222 | 75.88832 | 80.58839 | 75.88832 | 71.46228 |
| HOG SMOTE fold-wise | 100 | 100 | 100 | 100 | 85.50523 | 85.50703 | 85.50523 | 85.23506 | 76.14213 | 80.5422 | 76.14213 | 71.61638 |
| HOG + GLCM | 100 | 100 | 100 | 100 | 86.09756 | 86.14857 | 86.09756 | 85.83946 | 77.66497 | 81.59253 | 77.66497 | 73.08517 |
| HOG + GLCM SMOTE | 100 | 100 | 100 | 100 | 89.50967 | 89.48095 | 89.50967 | 89.29827 | 75.63452 | 81.17704 | 75.63452 | 71.18523 |
| HOG + GLCM SMOTE fold-wise | 100 | 100 | 100 | 100 | 86.27178 | 86.2235 | 86.27178 | 86.00244 | 77.15736 | 81.60183 | 77.15736 | 72.3263 |

**Table 9:** Summary results of the SVM classifier for Dataset

| SVM | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Feature type** | **Train accuracy** | **Train precision** | **Train recall** | **Train F1** | **Val accuracy** | **Val precision** | **Val recall** | **Val F1** | **Test accuracy** | **Test precision** | **Test recall** | **Test F1** |
| GLCM | 88.09233 | 88.22698 | 88.09233 | 88.10164 | 79.19861 | 79.2555 | 79.19861 | 79.11999 | 58.12183 | 56.37092 | 58.12183 | 55.21411 |
| GLCM SMOTE | 89.55562 | 89.56849 | 89.55562 | 89.54205 | 82.13386 | 82.10759 | 82.13386 | 82.06929 | 60.6599 | 60.47654 | 60.6599 | 57.40041 |
| GLCM SMOTE fold-wise | 88.15331 | 88.18451 | 88.15331 | 88.13782 | 79.23345 | 79.23173 | 79.23345 | 79.1685 | 58.88325 | 57.9985 | 58.88325 | 55.90959 |
| HOG | 100 | 100 | 100 | 100 | 86.41115 | 86.35051 | 86.41115 | 86.32499 | 77.15736 | 82.52629 | 77.15736 | 72.62883 |
| HOG SMOTE | 100 | 100 | 100 | 100 | 88.51223 | 88.40176 | 88.51223 | 88.4074 | 77.15736 | 82.64786 | 77.15736 | 72.44574 |
| HOG SMOTE fold-wise | 100 | 100 | 100 | 100 | 86.37631 | 86.29744 | 86.37631 | 86.28679 | 77.15736 | 82.60534 | 77.15736 | 72.65735 |
| HOG + GLCM | 100 | 100 | 100 | 100 | 89.44251 | 89.44691 | 89.44251 | 89.4085 | 75.88832 | 80.98018 | 75.88832 | 71.1002 |
| HOG + GLCM SMOTE | 100 | 100 | 100 | 100 | 91.35426 | 91.40164 | 91.35426 | 91.33736 | 76.14213 | 81.29979 | 76.14213 | 71.34485 |
| HOG + GLCM SMOTE fold-wise | 100 | 100 | 100 | 100 | 89.44251 | 89.46707 | 89.44251 | 89.42519 | 75.88832 | 81.12774 | 75.88832 | 71.13441 |

**Table 10:** Summary results of the LR classifier for Dataset 1

| LR | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Feature type** | **Train accuracy** | **Train precision** | **Train recall** | **Train F1** | **Val accuracy** | **Val precision** | **Val recall** | **Val F1** | **Test accuracy** | **Test precision** | **Test recall** | **Test F1** |
| GLCM | 75.4878 | 75.43806 | 75.4878 | 75.29089 | 73.0662 | 73.05491 | 73.0662 | 72.86605 | 44.67005 | 44.56164 | 44.67005 | 43.65774 |
| GLCM SMOTE | 76.19407 | 76.06494 | 76.19407 | 76.06351 | 74.12371 | 74.12552 | 74.12371 | 73.99889 | 36.80203 | 38.23625 | 36.80203 | 35.30616 |
| GLCM SMOTE fold-wise | 75.06098 | 75.08546 | 75.06098 | 74.93786 | 72.26481 | 72.39198 | 72.26481 | 72.15784 | 39.59391 | 40.72219 | 39.59391 | 38.02973 |
| HOG | 78.25784 | 77.92759 | 78.25784 | 77.99736 | 71.39373 | 70.90401 | 71.39373 | 71.01357 | 47.71574 | 50.63918 | 47.71574 | 45.87528 |
| HOG SMOTE | 80.10882 | 79.6599 | 80.10882 | 79.77071 | 74.12375 | 73.61798 | 74.12375 | 73.77273 | 48.22335 | 51.87898 | 48.22335 | 44.70792 |
| HOG SMOTE fold-wise | 78.02265 | 77.79131 | 78.02265 | 77.6994 | 71.14983 | 70.7939 | 71.14983 | 70.76528 | 48.73096 | 52.26121 | 48.73096 | 45.28948 |
| HOG + GLCM | 86.84669 | 86.82692 | 86.84669 | 86.80639 | 79.23345 | 79.14536 | 79.23345 | 79.15001 | 54.82234 | 54.98464 | 54.82234 | 52.43707 |
| HOG + GLCM SMOTE | 87.89302 | 87.78239 | 87.89302 | 87.81305 | 81.16691 | 80.97663 | 81.16691 | 81.02435 | 55.58376 | 56.78848 | 55.58376 | 52.72022 |
| HOG + GLCM SMOTE fold-wise | 86.6899 | 86.64837 | 86.6899 | 86.61685 | 78.15331 | 78.28142 | 78.15331 | 78.11855 | 56.59898 | 59.1702 | 56.59898 | 53.4943 |

As shown in Table 8, the K-Nearest Neighbours (KNN) classifier performed well across all feature types. However, the validating performance provides a more nuanced perspective, particularly in the context of SMOTE augmentation.

For texture-based total features (GLCM), the basic setup produced a test F1-score of 75.47 %, with a validation F1-score reaching 75.20 %, thereby creating a good benchmark. When the SMOTE was implemented globally, the validation performance improved to 79.13% F1, demonstrating more consistency among minority and majority classes in training. However, the test F1-rating declined significantly to 72.69%, indicating that oversampling may have caused some overfitting or feature space distortion. Notably, the fold-wise SMOTE lowered the test F1 to 70% while maintaining validation ranks near the baseline. These findings suggest that the SMOTE provided a modest benefit only on the validation F1-score, with a limited generalization advantage.

Shape-oriented HOG features, on the other hand, performed better across the board. The baseline HOG setup achieved a test F1 rating of 72.56%, whereas the validation F1 was at 84.75%, illustrating the intrinsic discriminative strength of gradient-based total descriptors. With SMOTE, the validation total performance improved by 87.52%, even though the test F1 fell somewhat to 71. 46%, demonstrating a capacity overfitting trend. On the other hand, the fold-wise SMOTE performed better, reaching 71.61% for the test F1, showing improved generalization when compared to the global SMOTE. These designs support the notion that the HOG features respond well to oversampling, also only on the validation set.

The blended HOG GLCM features space provided the best validation F1-ratings across all settings, with the SMOTE-augmented model scoring 89.29% and the fold-wise version scoring 86.00%. Interestingly, the test F1 for the mixed baseline was 73.08%, which decreased marginally with SMOTE fold-smart (72.32%). However, the normal SMOTE significantly reduced the overall performance to 71.18%. Despite the high validation scores, our findings suggest that the SMOTE yields diminishing results in complicated feature spaces where class impediments may already be well-described.

Overall, the results reveal that KNN maintained strong validation performance throughout all settings; however, the impact of SMOTE on test generalization became combined. While the SMOTE generally improved testing and validation balance, it did not consistently improve the test F1-rankings, demonstrating that the KNN's sensitivity to the distributional properties of oversampled statistics can limit its generalization under artificial augmentation, particularly in high-dimensional MRI feature areas.

As shown in Table 9**,** the SVM classifier demonstrated strong performance across various feature units, particularly using shape-based and blended descriptors. When using the GLCM features alone, the model demonstrated mild generalization capabilities, with a validation F1-score of 79.11% and a test F1-score of 55.21%. While the use of SMOTE increased the validation F1 to 82.06%, the test F1 score increased to 57.40%, indicating an influence on real-world generalization. The fold-wise SMOTE produced comparable validation overall performance (79.16%) and a slightly improved test F1-score (55.91%), confirming that class balance improved generalization slightly. These results confirm that, even if the GLCM features on their own are limited in separability, oversampling provides a measurable advantage.

In conclusion, the HOG features resulted in more powerful and consistent performance. Without augmentation, the SVM achieved a validation F1-score of 86.32% and a test F1 of 72.63%, indicating a significant improvement over the GLCM. In contrast, the SMOTE only slightly increased validation to 88.40%, with a high test F1 (72.44%). The fold-wise variation provided the best balance, with a validation F1 of 86.28% and the highest test F1-score in this configuration is 72.65%, reflecting a slight but significant advantage in generality. At the same time, augmentation is implemented with a go-verified balance. These data support the notion that SMOTE is more effective when the underlying capabilities already provide strong separability, as in the case of HOG.

The combination of HOG + GLCM feature set produced the best results for the length of validation, with the SMOTE increasing validation F1 to 91.33% and the fold-wise SMOTE remaining at 89.42%. In terms of the overall test f1-score performance, both SMOTE and fold-sensible SMOTE improved slightly but consistently, reaching 71.34% and 71.13%, respectively, after it was 71.10% without augmentation. Although these upgrades are not large, they indicate that even for rather discriminative mixed functions, the SMOTE can contribute to marginal test overall performance benefits, implying higher robustness under modest class imbalance conditions.

In summary, the SMOTE showed a positive influence on the test F1-scores across all feature types, with the most suggested advantage discovered in the GLCM (from 55.21 to 57.40%) and the highest average improvement seen in the HOG fold-wise SMOTE (72.65%). These data demonstrate that a magnificent balance via the SMOTE produces measurable improvements in generalization, particularly when combined with strong characteristic representations.

As seen in Table 10, the Logistic Regression classifier consistently underperformed SVM and KNN due to its linear character and poor capacity to describe complicated nonlinear patterns determined by brain tumor MRI information. Nonetheless, positive patterns appeared when examining the impact of the SMOTE augmentation on generalization, particularly using the Test F1 metric.

When the LR was used alone with the GLCM texture descriptors, it produced a validation F1-score of 72.86%. Still, its generalization plummeted dramatically, with a test F1-score of only 43.65%, indicating the model's limited robustness in real-world conditions. Interestingly, even though the SMOTE improved training and validation measures marginally, it did not result in a step forward in the overall performance, with the test F1-score dropping further to 35.31%. In contrast, the fold-wise SMOTE demonstrated a small recovery compared to the SMOTE.

On the other hand, the HOG features resulted in greater performance, which was more consistent with the LR's linear assumptions. The baseline HOG configuration achieved a validation F1 of 71.01% and a test F1 of 45.87%. After applying the SMOTE, the validation F1 increased to 73.77%, while the test F1 stayed practically steady at 44.70%, showing that there is no benefit from using it. However, the fold-wise SMOTE resulted in consistency in the test F1 of 45.29%.

The combination of HOG+ GLCM feature arrangement produced the greatest ordinary ratings for the LR. Without augmentation, it achieved 79.15% validation F1 and a test F1 of 52.43%. In particular, applying the SMOTE resulted in a minor improvement in both validation (81.02%) and examination (52.72%) F1-scores. Notably, the fold-sensible SMOTE provided the highest-quality test F1 across all LR configurations, hitting 53.49%, with a corresponding validation F1 of 78.11%. This outcome demonstrates that, while the SMOTE has a minor impact on the LR, the fold-sensible augmentation is slightly more powerful in constructing generalizable models when using wealthy, mixed capabilities.

In summary, the LR benefited marginally from the SMOTE, particularly in the fold-sensible setting, and typically when employed with the HOG or the hybrid functions. The highest improvement in the test F1 was observed in the HOG GLCM fold-wise setup (1.06 points above baseline). However, the total F1-score remained lower than those achieved by more complex classifiers, reinforcing that linear models such as LR are far less appropriate for brain tumor classification tasks, even though class balancing methods were used.

As seen in Table 11, the KNN classifier had consistently high performance across all feature types and configurations, displaying good generalization and benefiting greatly from the well-separated feature spaces. When the GLCM features were used alone, the classifier demonstrated good generalization, with a validation F1-score of 84.04% and a test F1-score of 87.47%. In particular, applying the SMOTE barely improved the validation performance to 87.34%, but the test F1-score remained at 87.34%, showing a minimal realistic impact. Interestingly, the fold-wise SMOTE maintained equal validation overall performance (84.24%) and obtained the highest test F1-score in this situation (87.74%), indicating the resilience of the GLCM features for the KNN, which also increased significantly with augmentation using the SMOTE fold-wise.

In comparison, the HOG capabilities produced substantially superior results. Without any augmentation, the model achieved a validation F1-score of 89.24% and an excellent test F1-score of 94.77%, demonstrating the HOG's discriminative energy in

KNN classification. The SMOTE increased validation to 91.80%, while the test F1 declined somewhat to 93.9%. The fold-smart technique achieved comparable validation (89.49%) and a test F1-score (93.86%), suggesting that for well-established features like HOG, augmentation provides minor benefits and can even lead to a significant reduction in generalization due to synthetic redundancy.

The blended feature set HOG + GLCM produced the best results, with the test F1-scores reaching 95.71% without augmentation. The SMOTE and the fold-wise SMOTE produced stable effects of 95.42%, matching the high performance of the initial feature set but failing to improve comparably. This stability demonstrates that the integrated functions currently provide sufficient class separability, with an enlargement that adds a little extra value.

In conclusion, the KNN benefited the most from the rich descriptors, such as HOG, with test-time performance already approaching the ideal case, particularly when paired with the GLCM. While the SMOTE occasionally increased validation scores, its impact on the test F1-scores was minor, especially when the baseline performance was already strong. However, the SMOTE improved the GLCM significantly (0.27 %).

**Table 11:** Summary results of the KNN classifier for Dataset 2

| KNN | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Feature type** | **Train accuracy** | **Train precision** | **Train recall** | **Train F1** | **Val accuracy** | **Val precision** | **Val recall** | **Val F1** | **Test accuracy** | **Test Precision** | **Test recall** | **Test F1** |
| GLCM | 100 | 100 | 100 | 100 | 84.24369 | 84.40324 | 84.24369 | 84.04362 | 87.79558 | 88.08949 | 87.79558 | 87.47723 |
| GLCM SMOTE | 100 | 100 | 100 | 100 | 87.41379 | 87.49145 | 87.41379 | 87.34781 | 87.64302 | 87.92599 | 87.64302 | 87.34003 |
| GLCM SMOTE fold-wise | 100 | 100 | 100 | 100 | 84.34894 | 84.42603 | 84.34894 | 84.24704 | 87.94813 | 88.05555 | 87.94813 | 87.74581 |
| HOG | 100 | 100 | 100 | 100 | 89.42576 | 89.49412 | 89.42576 | 89.24734 | 94.81312 | 94.84302 | 94.81312 | 94.77004 |
| HOG SMOTE | 100 | 100 | 100 | 100 | 91.8652 | 91.8972 | 91.8652 | 91.80759 | 93.97407 | 93.94469 | 93.97407 | 93.93985 |
| HOG SMOTE fold-wise | 100 | 100 | 100 | 100 | 89.60117 | 89.59598 | 89.60117 | 89.4962 | 93.89779 | 93.85746 | 93.89779 | 93.86974 |
| HOG + GLCM | 100 | 100 | 100 | 100 | 90.72126 | 90.75703 | 90.72126 | 90.63029 | 95.72845 | 95.74199 | 95.72845 | 95.71317 |
| HOG + GLCM SMOTE | 100 | 100 | 100 | 100 | 92.78997 | 92.82118 | 92.78997 | 92.75296 | 95.42334 | 95.42427 | 95.42334 | 95.42061 |
| HOG + GLCM SMOTE fold-wise | 100 | 100 | 100 | 100 | 91.14146 | 91.1625 | 91.14146 | 91.09086 | 95.42334 | 95.42427 | 95.42334 | 95.42061 |

As shown in Table 12, the SVM classifier demonstrated a great overall performance across all feature types, particularly the high training indicators, which regularly achieved optimal rankings. The effect of the SMOTE varied depending on the feature set. However, in most cases, it resulted in moderate increases or stability in the test-time performance.

With GLCM features, the baseline validation and the test F1-score were already high, at 87.91% and 90.07%, respectively. Applying the SMOTE resulted in a tiny increase in the validation F1 to 89.96% and a modest increase in the test F1 to 90.46%, indicating a minor advantage from balanced class representation. Interestingly, the SMOTE fold-wise approach caused the overall validation performance to decline somewhat (87.61%). Still, it achieved the highest test F1 on this group: 90.74%, indicating more potent generalization. At the same time, artificial data were restricted to the training folds, which supports the idea that the fold-wise augmentation can reduce overfitting to synthetic data while improving class balance.

Switching to the HOG features improved the overall performance on average. The baseline test F1-score increased to 92.86%, indicating the excessive separability of HOG features for SVM. The SMOTE increased the validation F1 to 92.18%, and the test F1 increased slightly to 93.09%, demonstrating a tiny but considerable benefit from oversampling. In testing, the fold-wise SMOTE provided nearly similar validation (89.29%) and minimally affected test overall performance (92.3%), implying that the entire training augmentation can be more potent when such powerful features are used.

The SVM classifier performed satisfactorily in terms of the mixed HOG + GLCM features. Without augmentation, the model achieved a test F1-score of 95.95%, which was nearly identical when the SMOTE (95.87%) or the SMOTE fold-wise(95.87%) was used. While validation ratings increased slightly with SMOTE (from 92.6% to 94.41%) and increased further (92.86%), test rankings remained stable, indicating that the rich joint descriptor area already provides sufficient separability and that SMOTE provides little additional generalization benefit in this example.

In summary, the SVM produced consistently solid results across all setups. The SMOTE, in both full and fold-wise modes, provided significant profits for the GLCM and reasonable improvements for HOG in each validation and test F1 scores. The impact on the mixed HOG GLCM features was low due to the strong baseline performance. Overall, the SMOTE performed best for single-descriptor feature areas, with the fold-wise augmentation occasionally providing pleasing generalization, particularly for texture-based features such as the GLCM.

**Table 12:** Summary results of the SVM classifier for Dataset 2

| SVM | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Feature type** | **Train accuracy** | **Train precision** | **Train recall** | **Train F1** | **Val accuracy** | **Val precision** | **Val recall** | **Val F1** | **Test accuracy** | **Test precision** | **Test recall** | **Test F1** |
| GLCM | 96.85749 | 96.92568 | 96.85749 | 96.85622 | 87.97278 | 88.0263 | 87.97278 | 87.91739 | 90.16018 | 90.30714 | 90.16018 | 90.0738 |
| GLCM SMOTE | 99.94514 | 99.94522 | 99.94514 | 99.94514 | 89.96865 | 89.97625 | 89.96865 | 89.96278 | 90.54157 | 90.57256 | 90.54157 | 90.46333 |
| GLCM SMOTE fold-wise | 99.96061 | 99.96067 | 99.96061 | 99.96061 | 87.67515 | 87.6028 | 87.67515 | 87.61003 | 90.84668 | 90.76805 | 90.84668 | 90.74338 |
| HOG | 100 | 100 | 100 | 100 | 89.42602 | 89.3941 | 89.42602 | 89.38431 | 92.90618 | 92.85055 | 92.90618 | 92.86799 |
| HOG SMOTE | 100 | 100 | 100 | 100 | 92.19436 | 92.19164 | 92.19436 | 92.18789 | 93.13501 | 93.08179 | 93.13501 | 93.098 |
| HOG SMOTE fold-wise | 100 | 100 | 100 | 100 | 89.33839 | 89.28061 | 89.33839 | 89.29015 | 92.67735 | 92.6135 | 92.67735 | 92.63424 |
| HOG + GLCM | 100 | 100 | 100 | 100 | 92.68218 | 92.68244 | 92.68218 | 92.66413 | 95.95728 | 95.99057 | 95.95728 | 95.95121 |
| HOG + GLCM SMOTE | 100 | 100 | 100 | 100 | 94.40439 | 94.46668 | 94.40439 | 94.41583 | 95.88101 | 95.90598 | 95.88101 | 95.87364 |
| HOG + GLCM SMOTE fold-wise | 100 | 100 | 100 | 100 | 92.85718 | 92.9236 | 92.85718 | 92.86145 | 95.88101 | 95.90598 | 95.88101 | 95.87364 |

**Table 13:** Summary results of the LR classifier for Dataset 2

| LR | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Feature type** | **Train accuracy** | **Train precision** | **Train recall** | **Train F1** | **Val accuracy** | **Val precision** | **Val recall** | **Val F1** | **Test accuracy** | **Test precision** | **Test recall** | **Test F1** |
| GLCM | 75.72217 | 75.7616 | 75.72217 | 75.64494 | 74.59744 | 74.67793 | 74.59744 | 74.50256 | 70.09916 | 69.87538 | 70.09916 | 69.85805 |
| GLCM SMOTE | 76.18339 | 76.26719 | 76.18339 | 76.11612 | 75.01567 | 75.13401 | 75.01567 | 74.94506 | 70.6331 | 70.52933 | 70.6331 | 70.49366 |
| GLCM SMOTE fold-wise | 75.6565 | 75.82293 | 75.6565 | 75.59566 | 74.545 | 74.70953 | 74.545 | 74.47694 | 70.78566 | 70.65305 | 70.78566 | 70.61067 |
| HOG | 80.79921 | 80.38649 | 80.79921 | 80.47192 | 77.64412 | 77.22105 | 77.64412 | 77.31949 | 73.76049 | 72.17294 | 73.76049 | 72.45914 |
| HOG SMOTE | 80.46238 | 80.12948 | 80.46238 | 80.205 | 77.46082 | 77.07296 | 77.46082 | 77.15483 | 72.61632 | 71.2649 | 72.61632 | 71.60271 |
| HOG SMOTE fold-wise | 80.89986 | 80.62558 | 80.89986 | 80.68201 | 77.04832 | 76.71602 | 77.04832 | 76.78203 | 72.61632 | 71.36687 | 72.61632 | 71.75159 |
| HOG + GLCM | 89.2857 | 89.21356 | 89.2857 | 89.22922 | 84.38383 | 84.33566 | 84.38383 | 84.31706 | 82.07475 | 81.78229 | 82.07475 | 81.85017 |
| HOG + GLCM SMOTE | 89.41223 | 89.38425 | 89.41223 | 89.38611 | 85.04702 | 85.0225 | 85.04702 | 85.0014 | 82.30359 | 82.10381 | 82.30359 | 82.11576 |
| HOG + GLCM SMOTE fold-wise | 89.21132 | 89.18796 | 89.21132 | 89.18638 | 83.87613 | 83.83651 | 83.87613 | 83.83397 | 82.45614 | 82.25598 | 82.45614 | 82.26794 |

As shown in Table 13, the LR classifier performed moderately and more variably than the other models, with training metrics well below the top rankings, indicating a reduced possibility for complicated decision restrictions. The effect of the SMOTE became dependent on the feature set, typically leading to tiny but consistent gains in real-time overall performance, particularly for weaker descriptors.

With GLCM features, the baseline validation and the test F1-score were 74.50% and 69.86%, respectively. Applying the SMOTE increased the validation F1 to 74.94% and the test F1 to 70.49%, demonstrating a moderate effect of balancing training. The fold-wise version had comparable validation performance (74.47%), but the best test F1 on this set was 70.61%, indicating significantly more potent generalization when artificial samples were limited to training folds.

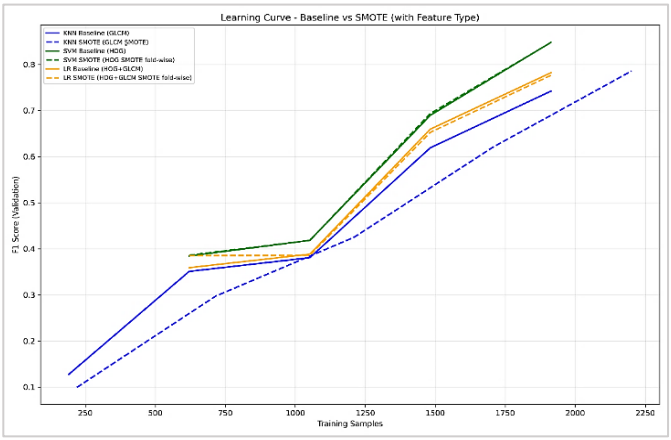
Using HOG features resulted in better results overall. The baseline F1-score was 72.45%, whereas the SMOTE maintained comparable levels (71.60%) with very minor changes in validation criteria. The fold-wise SMOTE technique achieved a slight decrease in F1 to 71.75%, demonstrating consistent behaviour but limited gain while the features are already quite discriminative.

In contrast, the LR delivered excellent results for the combined HOG GLCM functions. Without augmentation, the test F1-score reached 81.5 percent. The SMOTE increased the validation F1 from 84.31 to 85.00% and the test F1-score from 81.85 to 82.11%. In comparison, the fold-smart variant provided the highest test F1 at 82.26%, highlighting the ability of fold-wise augmentation to improve generalization without overfitting to artificial data.

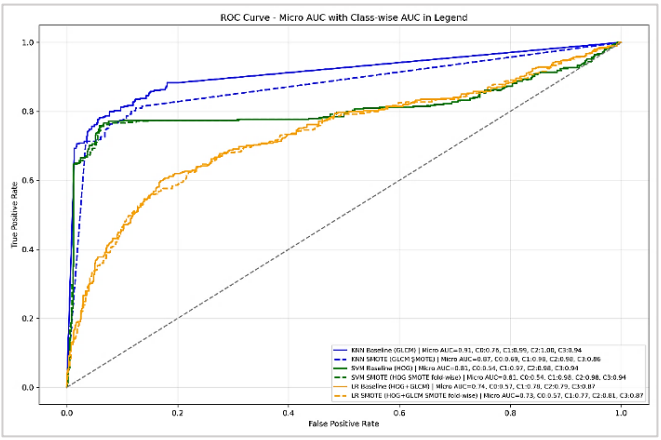
In conclusion, the LR profited the most from the SMOTE when using considerably fewer discriminative functions, such as GLCM, with modest improvements for more powerful feature units. The fold-wise technique produced excellent test-time performance, reinforcing its usefulness for improving generalization in decrease-capacity models.

The Receiver Operating Characteristic (ROC) curve and its associated Area Under the Curve (AUC) provide a comprehensive evaluation of the classifier's ability to distinguish between classes. These curves illustrate the trade-off between False Positive Rate (FPR) and True Positive Rate (TPR), with micro AUC and per-class AUC values. The Classes C0, C1, C2, and C3 correspond to glioma, meningioma, no tumor, and pituitary, respectively. This improvement can be due to the balanced distribution of instructions completed using SMOTE. The AUC values after augmentation are minimally increased in some feature extraction techniques (such as HOG in KNN and HOG + GLCM in LR), and these results are consistent with the previous F1-score analysis findings, indicating that the SMOTE adds to magnificent balance and slight robustness improvements, albeit not uniformly across all models. As shown in Figure 6 a, the learning curves demonstrate the training and validation performance of the classifiers, while Figure 6b presents the ROC curves highlighting the improvements achieved through SMOTE augmentation.

For the second dataset, we analyzed each classifier's baseline setup and the most advanced SMOTE-based configuration (not the best method with SMOTE across all methods in the single classifier). Specifically, for the KNN, the baseline became HOG + GLCM, and the improved one was the GLCM with SMOTE (fold-wise); for SVM, the baseline became HOG + GLCM, and the improved one became HOG with SMOTE; and for the LR, the baseline and improved results were acquired using HOG + GLCM and HOG + GLCM SMOTE fold-wise, which is the most beneficial from the SMOTE. For the second dataset, we analyzed each classifier's baseline setup and the most advanced SMOTE-based configuration (not the best method with SMOTE across all methods in the single classifier). Specifically, for the KNN, the baseline became HOG + GLCM, and the improved one was GLCM with SMOTE (fold-wise); for the SVM, the baseline became HOG + GLCM, and the improved one was HOG with SMOTE; and for the LR, the baseline and the improved results were acquired using HOG + GLCM and HOG + GLCM SMOTE fold-wise, which is the most beneficial from the SMOTE. As shown in Figure 7a, the learning curves demonstrate the differences between baseline and SMOTE-augmented configurations for each classifier. Figure 7b presents the corresponding ROC curves, which confirm the improvements obtained with SMOTE augmentation.

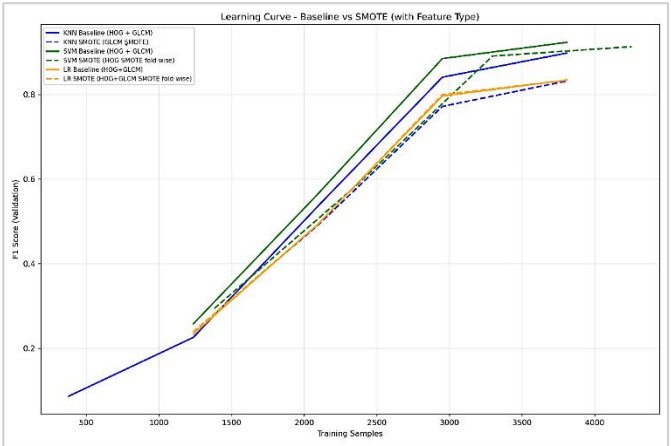



**(a)**

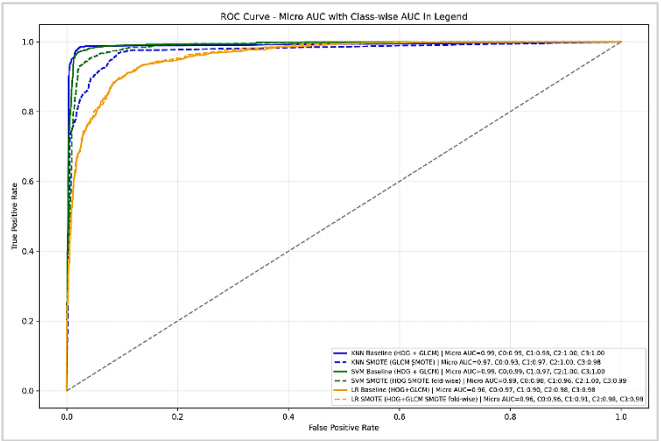



**(b)**

**Figure 6:** a) Learning curves and b) ROC curves for the first dataset, illustrating the configurations under baseline and with SMOTE augmentation: KNN (GLCM features), SVM (HOG features), and Logistic Regression (HOG + GLCM features), SMOTE settings




**(a)**




**(b)**

**Figure 7:** a) Learning curves and b) ROC curves for the second dataset, illustrating the configurations for KNN: HOG + GLCM, GLCM SMOTE; for SVM: HOG + GLCM, HOG SMOTE fold-wise; and for Logistic Regression (LR): HOG + GLCM, HOG + GLCM SMOTE fold-wise

## 5. Conclusion

This study examined the effect of SMOTE augmentation, applied in both complete and fold-wise modes, on brain tumor MRI classification using hand-made features (GLCM, HOG, and HOG GLCM) and three classic classifiers (KNN, SVM, and LR) across two datasets.

For Dataset 1, the most significant improvement was reported for SVM with GLCM features, with the Test F1-score increasing from 55.21 (baseline) to 57.40% with the complete SMOTE and 55.91% with the fold-wise SMOTE. The LR gained the greatest advantage with HOG GLCM, increasing from 52.4 to 53.5%. However, the KNN revealed limited interchange, indicating a lower sensitivity to oversampling on this dataset.

For Dataset 2, the fold-wise SMOTE improved the GLCM performance in KNN from 87.48 to 87.75%, while the SVM profited significantly in GLCM (90.07% to 90.74%) and HOG (92.87% to 93.10%) compared to the full SMOTE. The LR made significant gains with GLCM (69.85% - 70.61%) and HOG + GLCM (81.85% - 82.26%) by utilizing the fold-wise technique. The combined HOG GLCM features in SVM maintained a great baseline overall performance (~95.9%) with negligible changes, indicating that the SMOTE is less effective when characteristic richness is already high.

Overall, the findings suggest that the effectiveness of SMOTE is substantially influenced by feature discriminability, classifier type, and augmentation strategy. The fold-wise variation generally improved generalization by preventing information leakage, whereas the complete SMOTE occasionally offered higher validation scores but carried a minimal risk of overfitting. These findings support SMOTE as a valuable tool for improving performance in imbalanced medical imaging datasets, particularly for weaker texture-based descriptors such as the GLCM.

### Author contributions

Conceptualization **F. Fadhil** and **Z. Sultani**; data curation, **F. Fadhil**; formal analysis, **F. Fadhil**; investigation, **F. Fadhil**; methodology, **F. Fadhil**; project administration, **Z. Sultani**; resources, **F. Fadhil**; software, **F. Fadhil**; supervision, **Z. Sultani**; validation, **Z. Sultani** and **F. Fadhil**; visualization, **F. Fadhil**; writing—original draft preparation, **F. Fadhil**; writing—review and editing, **Z. Sultani**. All authors have read and agreed to the published version of the manuscript..

### Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

### Data availability statement

The data that support the findings of this study are available on request from the corresponding author.

### Conflicts of interest

The authors declare that there is no conflict of interest.

## References

**[1]** M. Azeez Joodi, M. Hadi Saleh, D. Jasim Kadhim, A New Proposed Hybrid Learning Approach with Features for Extraction of Image Classification, J. Robotics, 2023 (2023)1-13. https://doi.org/10.1155/2023/9961421

**[2]** J. Chen, Y. Zeng, Application of machine learning in rock facies classification with physics-motivated feature augmentation, arXiv preprint arXiv:1808 (2018) 09856. https://doi.org/10.48550/arXiv.1808.09856

**[3]** J. Rama, C. Nalini, A. Kumaravel, Image pre-processing: enhance the performance of medical image classification using various data augmentation technique, ACCENTS Transactions on Image Processing and Computer Vision, 5 (2015) 7-14. http://dx.doi.org/10.19101/TIPCV.2018.413001

**[4]** D. A. Dablain, N. V. Chawla, Towards understanding how data augmentation works with imbalanced data, arXiv preprint arXiv, 2304 (2023) 05895. https://doi.org/10.48550/arXiv.2304.05895

**[5]** K. Alomar, H. I. Aysel, X. Cai, Data augmentation in classification and segmentation: A survey and new strategies, J. Imaging, 9 (2023) 46. https://doi.org/10.3390/jimaging9020046

**[6]** Kalaivani, S., Asha, N., Gayathri A. 2023. Geometric transformations-based medical image augmentation, InGANs for Data Augmentation in Healthcare, Cham: Springer International Publishing, pp. 133–141. https://doi.org/10.1007/978-3-031-43205-7_8

**[7]** J. Liu, Importance-SMOTE: a synthetic minority oversampling method for noisy imbalanced data, Soft Comput., 26 (2022) 1141–1163. https://doi.org/10.1007/s00500-021-06532-4

**[8]** Y. Wang, Y. Ji, H. Xiao, A data augmentation method for fully automatic brain tumor segmentation, Comput. Biol. Med., 149 (2022) 106039. https://doi.org/10.1016/j.compbiomed.2022.106039

**[9]** H. Wang, S. Tian, Y. Fu, J. Zhou, J. Liu, D. Chen, Feature augmentation based on information fusion rectification for few-shot image classification, Sci. Rep., 13 (2023) 3607. https://doi.org/10.1038/s41598-023-30398-1

**[10]** Y. Hasan, T. Khan, D. R. F. De Bulnes, J. F. H. Albarracin, C. Ryan, A Comparative Analysis of Implicit Augmentation Techniques for Breast Cancer Diagnosis Using Multiple Views, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2024, 2345-2354.

**[11]** M. Z. Alam, T. Roy, H. M. N. Kawsar, I. Rimi, Enhancing Transfer Learning for Medical Image Classification with SMOTE: A Comparative Study, 27th International Conference on Computer and Information Technology (ICCIT), Cox's Bazar, Bangladesh, 2024, 245-250. https://doi.org/10.1109/ICCIT64611.2024.11022326

**[12]** S. Harish, G. F. A. Ahammed, Integrated modelling approach for enhancing brain MRI with flexible pre-processing capability, Int. J. Electr. Comput. Eng., 9 (2019) 2416. http://doi.org/10.11591/ijece.v9i4.pp2416-2424

**[13]** Islam, M. A. Comparative analysis of pre-trained models and interpolation for facial expression recognition. M.Sc. Thesis, Metropolia University of Applied Sciences, 2023. https://www.theseus.fi/handle/10024/800196

**[14]** L. Dalavai, N. M. R. Purimetla, S. S. Vellela, T. SyamsundaraRao, L. R. Vuyyuru, K. K. Kumar, Improving Deep Learning-Based Image Classification Through Noise Reduction and Feature Enhancement, International Conference on Artificial Intelligence and Quantum Computation-Based Sensor Application (ICAIQSA), Nagpur, India, 2024, 1-7. https://doi.org/10.1109/ICAIQSA64000.2024.10882201

**[15]** Z. Rasheed, Y. K. Ma, I. Ullah, Y. Y. Ghadi, M. Z. Khan, M. A. Khan, A. Abdusalomov, F. Alqahtani, Brain tumor classification from MRI using image enhancement and convolutional neural network techniques, Brain Sci., 13 (2023) 1320. https://doi.org/10.3390/brainsci13091320

**[16]** I. M. Mohammed, N. A. M. Isa, Contrast Limited Adaptive Local Histogram Equalization Method for Poor Contrast Image Enhancement, IEEE Access, 13 (2025) 62600-62632. https://doi.org/10.1109/ACCESS.2025.3558506

**[17]** N. J. Wala'a, J. M. Rana, A survey on segmentation techniques for image processing, Iraqi J. Electr. Electron. Eng., 17 (2021) 73-93. http://ijeee.edu.iq/Papers/Vol17-Issue2/1570736047.pdf

**[18]** A. Kesana, J. Nallola, R. T. Bootapally, Brain Tumor Detection Using YOLOv5 and Faster R-CNN, 2nd International Conference on Vision Towards Emerging Trends in Communication and Networking Technologies (ViTECoN), Vellore, India, 2023, 1-6. https://doi.org/10.1109/ViTECoN58111.2023.10157773

**[19]** Cai, X., Li, X., Razmjooy, N. Breast cancer diagnosis by convolutional neural network and advanced thermal exchange optimization algorithm, Comput. Math. Methods Med., 2021 (2021) 1-13. https://doi.org/10.1155/2021/5595180

**[20]** M. Ahammed, M. Al Mamun, M. S. Uddin, A machine learning approach for skin disease detection and classification using image segmentation, Healthcare Analytics, 2 (2022) 100122. https://doi.org/10.1016/j.health.2022.100122

**[21]** M. Nazir, Z. Jan, M. Sajjad, Facial expression recognition using histogram of oriented gradients based transformed features, Cluster Comput., 21 (2018) 539-548. https://doi.org/10.1007/s10586-017-0921-5

**[22]** Y. Nizamli, A. Filatov, MRI brain tumor classification using HOG features selected via impurity-based importances measure, Int. J. Electr. Electron. Res., 12 (2024) 1251-1257. https://doi.org/10.37391/IJEER.120416

**[23]** S. Barburiceanu, R. Terebes, S. Meza, 3D texture feature extraction and classification using GLCM and LBP-based descriptors, Appl. Sci., 11 (2021) 2332. https://doi.org/10.3390/app11052332

**[24]** M. Shahajad, D. Gambhir, R. Gandhi, Features extraction for classification of brain tumor MRI images using support vector machine, 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 2021, 767-772. https://doi.org/10.1109/Confluence51648.2021.9377111

**[25]** F. T. Kurniati, D. H. F. Manongga, E. Sediyono, GLCM-based feature combination for extraction model optimization in object detection using machine learning, J. Ilm. Tek. Elektro Komput. Dan Inform., 9 (2023) 1196-1205. https://doi.org/10.26555/jiteki.v9i4.27842

**[26]** B. Pattanaik, K. Anitha, S. Rathore, P. Biswas, P. Sethy, S. Behera, Brain tumor magnetic resonance images classification based machine learning paradigms, Contemporary Oncology/Współczesna Onkologia, 26 (2022) 268-274. https://doi.org/10.5114/wo.2023.124612

**[27]** G. Dheepak, D. Vaishali, Brain tumor classification: a novel approach integrating GLCM, LBP and composite features, Front. Oncol., 13 (2024) 1248452. https://doi.org/10.3389/fonc.2023.1248452

**[28]** J. Wang, N. Awang, MKC-SMOTE: A Novel Synthetic Oversampling Method for Multi-Class Imbalanced Data Classification, IEEE Access, 12 (2024) 196929-196938. https://doi.org/10.1109/ACCESS.2024.3521120

**[29]** M. Z. Alam, T. Roy, H. M. N. Kawsar, I. Rimi, Enhancing transfer learning for medical image classification with smote: A comparative study, 27th International Conference on Computer and Information Technology (ICCIT), Cox's Bazar, Bangladesh, 2024, 245-250. https://doi.org/10.1109/ICCIT64611.2024.11022326

**[30]** F. R. Adi Pratama, S. I. Oktora, Synthetic Minority Over-sampling Technique (SMOTE) for handling imbalanced data in poverty classification, Stat. J. IAOS., 39 (2023) 233-239. https://doi.org/10.3233/SJI-220080

**[31]** N. Hameed, A. M. Shabut, M. K. Ghosh, M. A. Hossain, Multi-class multi-level classification algorithm for skin lesions classification using machine learning techniques, Expert Syst. Appl., 141 (2020) 112961. https://doi.org/10.1016/j.eswa.2019.112961

**[32]** S. K. Chauhan, B. Jaysawal, J. K. Bhalani, P. K. Sahoo, S. R. Parija, S. B. Shah, A. Thakur, Implementation and performance analysis of k-nearest neighbors algorithm for classification, in IET Conference Proceedings CP920. 2025. IET. https://doi.org/10.1049/icp.2025.1656

**[33]** S. N. Khan, S. U. Khan, H. Aznaoui, C. B. Şahin, Ö. B. Dinler, Generalization of linear and non-linear support vector machine in multiple fields: a review, Computer Science and Information Technologies, 4 (2023) 226-239. https://doi.org/10.11591/csit.v4i3.pp226-239

**[34]** J. Sultana, A. K. Jilani, Predicting breast cancer using logistic regression and multi-class classifiers, Int. J. Eng. Technol., 7 (2018) 22-26. https://doi.org/10.14419/ijet.v7i4.20.22115

**[35]** J. Zhang, X. Tan, W. Chen, G. Du, Q. Fu, H. Zhang, H. Jiang, EFF_D_SVM: a robust multi-type brain tumor classification system, Front. Neurosci., 17 (2023) 1269100. https://doi.org/10.3389/fnins.2023.1269100