Nigerian Research Journal of Engineering and Environmental Sciences 9(2) 2024 pp. 664-680
p ISSN: 2635-3342; e ISSN: 2635-3350

Nigerian Research Journal of Engineering and Environmental Sciences
Journal homepage: www.rjees.com

Original Research Article

Overcoming Computational Challenges of K-Nearest Neighbors: A Memory-Efficient Approach using Gray Level Co-Occurrence Matrix and Firefly Optimization Technique

***[1]Olusi, T., [2]Balogun, M.O., [3]Adepoju, M.T., [4]Sotonwa, K.A., [5]Adetunji, B., [5]Olabiyisiisi, O. and [3]Omidiora, O.**

*[1]Department of Computer Science, I. I. C. T. Kwara State Polytechnic, Ilorin, Nigeria.
[2]Department of Electrical and Computer Engineering Kwara State University, Malete, Kwara State Nigeria.
[3]Department of Computer Engineering, Ladoke Akintola University of Technology, Oyo State, Nigeria.
[4]Department of Computer Science, Lagos State University, Lagos, Nigeria.
[5]Department of Computer Science Ladoke Akintola University of Technology, Ogbomoso, Oyo State, Nigeria.
*olusititilayo@gmail.com; tmadepoju@lautech.edu.ng; monsurat.balogun@kwasu.edu.ng; kehinde.sotonwa@lasu.ng

http://doi.org/10.5281/zenodo.14565971

ARTICLE INFORMATION

Article history:
Received 24 Oct. 2024
Revised 02 Nov. 2024
Accepted 11 Nov. 2024
Available online 30 Dec. 2024

Keywords:
GLCM
Firefly
GLCM-Firefly-KNN
K-nearest neighbor
Feature extraction

ABSTRACT

Feature extraction is a critical step in machine learning, especially when dealing with high-dimensional datasets such as medical data. The k-Nearest Neighbors (KNN) algorithm is widely used for feature extraction due to its simplicity and effectiveness, but it often suffers from high memory consumption, particularly with large datasets such as breast cancer dataset. This work addresses the problem of optimizing memory efficiency of KNN, which can limit its applicability in resource-constrained environments, by proposing a novel approach through integrating the Gray Level Co-Occurrence Matrix (GLCM) for feature extraction with the Firefly optimization technique. GLCM is was employed to reduce the memory requirements of the dataset by extracting texture features, thus minimizing the computational cost during classification. The Firefly algorithm is was then utilized to optimize the selection of relevant features, further improving the memory efficiency and scalability of KNN. Datasets of over 1,000 locally collected breast cancer records were used. The datasets were partitioned into five folds to assess the performance of the models. The memory efficiency of standard KNN was compared with that of KNN optimized with the GLCM and Firefly algorithm (GLCM-FA-KNN). Results show that the GLCM-FA-KNN significantly outperformed the standard KNN in terms of memory efficiency, with an overall average of 5.31KB compared to 9.90KB for KNN. These findings demonstrate that GLCM-FA optimization improves KNN's memory usage, making it a more suitable choice for large-scale data applications.

1. INTRODUCTION

Feature extraction is a fundamental task in the development of machine learning models, as it directly impacts the efficiency and effectiveness of data classification, especially when dealing with high-dimensional datasets (Benyahia et al., 2022). In medical data analysis, where large volumes of complex information are involved such as breast cancer dataset, the need for efficient algorithms becomes critical (Almutlaq et al., 2023).

In modern times, numerous diseases contribute to mortality among women, with breast cancer being one of the most impactful (Labrada et al., 2021.). It is regarded as the most common form of cancer affecting women (Krishna et al., 2021). Data mining tools have been effectively applied in the diagnosis and classification of breast cancer, thereby addressing real-world medical challenges (Abdolrazzagh-Nezhad et al., 2020). Breast cancer typically begins with the uncontrolled growth of cells in the breast (Albert, 2023). As these cells multiply excessively, they form a tumor, which may be felt as a lump or detected through an X-ray (Barrientos et al., 2022). The cells become malignant and cancerous when they invade breast tissue or spread to other parts of the body (Hanker et al., 2020). There are several methods used to detect cancerous cells in the human body, ranging from imaging techniques to laboratory tests, biopsies, machine learning (ML) algorithms and so on (Park et al., 2022).

Machine learning (ML) algorithms have become pivotal in the detection of cancer cells, offering more efficient, accurate, and non-invasive approaches (Yadav et al., 2022). These algorithms can analyze large datasets from medical imaging, genetic data, and patient records to detect patterns that indicate the presence of cancerous cells, among which are Support Vector Machine (SVM), Artificial Neural Network (ANN), Random Forest (RF), Decision Tree (DT), Convolution Neural Network (CNN), K-Nearest Neighbor (KNN) and so on (Lopez et al., 2020).

In the field of data mining, KNN algorithms are highly popular due to their relatively quick learning process (Sabry, 2023). Among the various learning algorithms, KNN is considered one of the simplest to implement, as it operates on the principle that elements in proximity tend to share similar characteristics (Juna, et al., 2022). Once the features of one element are identified, predicting the features of other nearby elements becomes straightforward. KNN is derived from the nearest neighbor approach, where new instances are classified based on the majority vote or similarity of their 'K' neighbors with K being a small, positive integer (Ukey et al., 2023a). KNN is widely utilized due to its simplicity and ability to achieve high classification accuracy (Zhao et al., 2024).

However, its major drawback is its high memory consumption, particularly when working with large datasets (Zahirah et al., 2024). This challenge can make KNN impractical for resource-constrained environments or systems with limited computational power. In solving these limitations, optimization techniques have been introduced to enhance the performance of machine learning algorithms. Algorithms such as Principal Components Analysis (PCA), Linear Discriminant Analysis (LDA), Particle Swamp Optimization (PSO), Genetic Algorithm (GA) and Firefly Algorithm are among techniques that have been used to improve the performance of KNN (Kherif et al., 2021). PCA is a popular dimensionality reduction technique that projects high-dimensional data into a lower-dimensional space while preserving as much variance as possible (Rahman et al., 2020). LDA is a dimensionality reduction technique that focuses on maximizing the separation between different classes in the dataset (Peng et al., 2024). Genetic Algorithms (GA), Particle Swarm Optimization (PSO), and Firefly algorithm, have been used to select relevant features that improve the classification performance and reduce memory usage in KNN (Zhu et al., 2020). These algorithms search for optimal subsets of features that contribute to better classification accuracy while discarding irrelevant or redundant ones, thereby reducing the computational load of KNN and helping in reducing noise from irrelevant features (Applications, 2023).

Firefly algorithm is a nature-inspired metaheuristic approach based on the behavior of fireflies (Koosha et al., 2022). The Firefly algorithm is well known for its efficiency in solving optimization problems, including feature selection, where irrelevant or redundant features are discarded to improve the performance of models (Khan et al., 2021). By applying Firefly optimization to KNN, it becomes possible to not only maintain or improve classification accuracy but also significantly reduce memory usage.

Gray Level Co-Occurrence Matrix (GLCM) is a statistical method used in image processing to analyze the texture of an image (Sinaga et al., 2021). It provides a way to quantify how often pairs of pixel intensity values (gray levels) occur in a specific spatial relationship in an image. The GLCM is particularly useful in extracting texture features that describe the spatial distribution and patterns of pixel intensities in an image, making it valuable for tasks like texture classification, medical image analysis, and object detection (Hussain et al., 2022). Using the Gray Level Co-Occurrence Matrix (GLCM) to improve the memory usage of K-Nearest Neighbors (KNN) is a promising approach, especially when dealing with large image datasets where memory and computational efficiency are critical (Djunaidi et al., 2021). The idea is to use GLCM to extract meaningful texture features from images, reducing the dimensionality of the data that KNN needs to process. This approach leads to better memory usage and faster computations without sacrificing classification performance.

In this study, the memory efficiency of the standard KNN algorithm and a KNN model optimized using GLCM and Firefly algorithms were investigated. Using a dataset of One thousand, six hundred and five (1605) mammographic images, partitioned into five folds. The goal is to evaluate the extent to which the GLCM-Firefly-KNN can reduce memory consumption while maintaining or improving classification accuracy. The results indicate that the optimized GLCM-Firefly-KNN significantly outperforms the standard KNN in terms of memory efficiency, making it a more viable solution for large-scale, data-intensive applications such as medical diagnosis.

2. MATERIALS AND METHODS

This section outlines the methodology for developing a memory-efficient K-Nearest Neighbors (KNN) algorithm using the Gray Level Co-Occurrence Matrix (GLCM) for feature extraction and the Firefly Optimization Technique for further optimization of the KNN feature extraction. The methodology consists of four main stages: Data Collection and Preprocessing, relevant datasets are gathered and prepared for analysis at this stage. The next stage is features extraction using GLCM-FA-KNN algorithm, at this stage, the performance of KNN was optimized using Gray Level Co-Occurrence Matrix (GLCM) and Firefly Algorithm (FA), the optimized KNN was then used for features extraction. Results were evaluation to assess the performance of the model based on memory usage, accuracy, and other relevant metrics, this was followed by results analysis, which entails interpretation of findings by comparing the optimized model against baseline results to determine improvements and effectiveness. The methodology was structured into five major phases as shown in Figure 1.

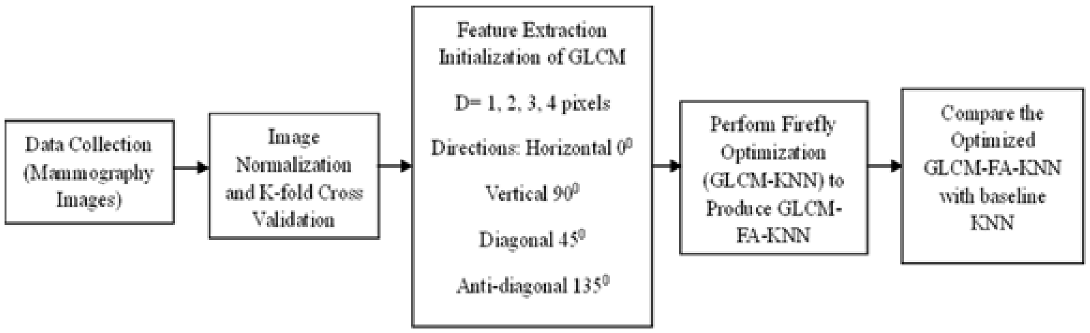
Data Collection (Mammography Images)
Image Normalization and K-fold Cross Validation
Feature Extraction Initialization of GLCM
D= 1, 2, 3, 4 pixels
Directions: Horizontal 0⁰
Vertical 90⁰
Diagonal 45⁰
Anti-diagonal 135⁰
Perform Firefly Optimization (GLCM-KNN) to Produce GLCM-FA-KNN
Compare the Optimized GLCM-FA-KNN with baseline KNN

Figure 1: Block diagram of the developed GLCM-FA-KNN algorithm

Phase 1: Data preparation and preprocessing:

- Collection and preprocessing of breast cancer image data (resize, normalize and so on).
- Splitting of the dataset into training and testing sets using 5-fold cross-validation.

Phase 2: Feature Extraction Using GLCM

- Computing GLCM for each image and extract texture features (contrast, energy, homogeneity, correlation).

Phase 3: KNN-FA Optimization

- Initialization of KNN algorithm with random parameters.
- Applying the Firefly Algorithm to optimize KNN's parameters based on classification accuracy and memory efficiency.

Phase 4: Compare the results of the proposed method (GLCM-FA-KNN) with baseline KNN.

Python Programming Environment with libraries Scikit-learn was used for KNN implementation, OpenCV for GLCM feature extraction, and custom implementations of the Firefly Algorithm were used. One thousand, six hundred and five (1605) local mammographic images were collected from the University of Ilorin Teaching Hospital sample of which is presented in Figure 2. The database consists of 620 Malignant, 553 Benign and 412 Normal. Twenty (20) irrelevant data were discarded from the dataset, since the centroid of mass was not specified in them. The discarded data were eight (8) Benign, seven (7) Malignant and five (5) of Normal type. Therefore, the total number of data images used for the system was one thousand five hundred and eighty-five (1585) digital mammograms. The acquired breast images were pre-processed for the purpose of enhancing the quality of data set by reducing irrelevant data that can interfere with the training process.

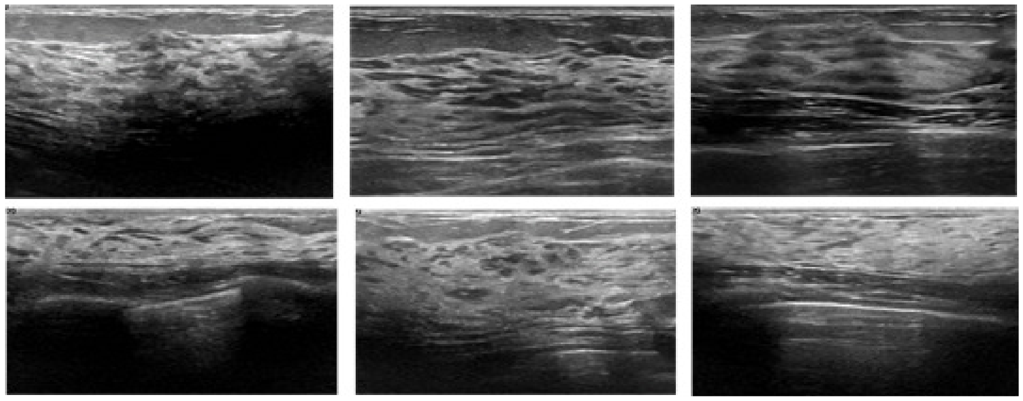
Figure 2: Sample of the collected mammographic images

The datasets were partitioned into training and testing sets using a 5-fold cross-validation approach. This ensures that the model's performance is robust and not overly biased by a particular training or testing set (Kaliappan *et al.*, 2023). Prior to feature extraction, the images were resized and normalized to ensure uniformity in pixel values, which will aid in consistent feature extraction and classification. Gray Level Co-Occurrence Matrix (GLCM) was used for feature extraction. GLCM was used to reduce the dimensionality of the raw image data by extracting key texture features that describe the spatial relationships between pixel intensities.

GLCM is widely used for feature extraction in image analysis (Riesaputri *et al.,* 2020). The GLCM is a matrix where the number of rows and columns equals the number of possible gray levels in the image. Each element(i, j) of the matrix represents the frequency with which two pixels, separated by a certain distance and angle, have intensity values i and j, respectively.

Let $p(i, j/d, \theta)$ represent the GLCM, where:

- i, j are the gray levels (intensity values) in the image.
- d is the distance between two pixels.
- θ is the angle or direction between the two pixels (for instance, 0°, 45°, 90°, 135°).

The element $p(i,j/d,\theta)$ represents how many times a pixel with gray level i and a pixel with gray level j occur at a distance d and direction θ . The GLCM was calculated using Equation 1.

$$p(i,j/d,\theta) = \sum_{x=1}^{N-dz} \sum_{y=0}^{M-d_y} \begin{cases} 1 \; if \; I(x.y) = i \; and \; I\big((x+d_x, y+d_y\big) \\ 0 \qquad\qquad otherwise \end{cases} \tag{1}$$

Where:

- N and M are the dimensions of the image.
- $I(x.y)$ is the intensity (gray level) at pixel (x,y).
- $d_x$and d_y are the horizontal and vertical distances, respectively, determined by d and θ.

For example, at $\theta = \theta^0,\ d_x = d$ and $d_y = 0$.

At θ=$90^0,\ d_x = 0$ and $d_y = d$.

To obtain a normalized GLCM, where the values represent probabilities instead of raw counts, each element of the GLCM was divided by the total number of pixel pairs as shown in Equation 2.

$$p'(i,j\backslash d,\theta) = \frac{p(i,j\backslash d,\theta)}{\sum_{i,j} p(i,j\backslash d,\theta)} \tag{2}$$

Where $\sum_{i,j} p(i,j\backslash d,\theta)$ is the sum of all the elements in the GLCM.

Once the GLCM is computed, several texture features can be extracted. The most common ones are:

Contrast which measures the intensity contrast between a pixel and its neighbor over the entire image. It is defined as shown in Equation 3.

$$Contrast = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} (i-j)^2 p'(i,j\backslash d,\theta) \tag{3}$$

Where G is the number of gray levels in the image.

Energy measures the uniformity or textural smoothness. It is the sum of squared elements in the GLCM as seen in Equation 4.

$$Energy = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} p'(i,j\backslash d,\theta)^2 \tag{4}$$

Homogeneity measures the closeness of the distribution of elements in the GLCM to the diagonal. It is higher for images with minimal variation in gray levels between neighboring pixels, the formula for homogeneity is shown in Equation 5.

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

Correlation measures how correlated a pixel is to its neighbor, considering the mean and variance of gray levels as shown in Equation 6.

$$Correlation = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} \frac{(i-\mu_i)(j-\mu_j)p'(i,j\backslash d,\theta)}{\sigma_i \sigma_j} \tag{6}$$

Where:

- μ_i and μ_j are the means of row and column intensities.
- σ_i and $\sigma_j$are the standard deviations of row and column intensities.

The mean and standard deviation for the intensity levels can be calculated using Equations 7 and 8.

$$\mu_i = \sum_{i=0}^{G-1} i \sum_{j=0}^{G-1} p'\,(i,j\backslash d,\theta) \tag{7}$$

$$\sigma_i = \sqrt{\sum_{i=0}^{G-1}(i-\mu_i)^2 \sum_{j=0}^{G-1} p(i,j\backslash d,\theta)} \quad (8)$$

The same equations apply for μ_i and σ_i by switching the summation indices.

Entropy measures the randomness of the gray level distribution. A higher entropy value corresponds to a more complex texture, this is determined using Equation 9.

$$Entropy = \sum_{i=0}^{G-1}\sum_{j=0}^{G-1} p'(i,j\backslash d,\theta) log_2(p'(i,j\backslash d,\theta)) \quad (9)$$

GLCM provides a comprehensive way to quantify texture by considering pixel relationships at different distances and directions. From the GLCM, various features like Contrast, Energy, Homogeneity, Correlation, and Entropy were extracted. These features are used for memory efficiency of KNN by reducing the dimensionality and focusing on key texture characteristics instead of raw pixel data.

Algorithm 1: Generating set of predefined offsets by GLCM

For each image, a GLCM calculated a set of predefined offsets and directions horizontal, vertical, and diagonal using Algorithm 1.

```
Begin
    For every image $X_{(i.j)}$
    Horizontal $(0^0)$
    Vertical $(90^0)$
    Diagonal: 3. Diagonal:
    Bottom left to top right $(-45^0)$
    Top left to bottom right $(-135^0)$
    Denoted $P_0$, $P_{45}$, $P_{90}$, and $P_{135}$ Respectively.
End.
```

Where, P_0 is the $PP_{(i,j)}$

For each image, a GLCM was calculated for a set of predefined offsets and directions (horizontal, vertical, and diagonal) as shown in Algorithm1. The GLCM was computed at different distances (1, 2, 4 pixels), capturing texture patterns at various scales. From the GLCM, statistical texture features were extracted, such as: the intensity contrast between neighboring pixels, energy which indicates the uniformity or smoothness of texture, homogeneity which measures the similarity of pixel intensities, Correlation for capturing the relationship between pixel intensities. The extracted features serve as input for the KNN classifier, significantly reducing the dimensionality of the data compared to using raw pixel values.

Optimizing GLCM-KNN using Firefly Algorithm

The Firefly Algorithm (FA) was used to optimize the parameters of the KNN algorithm, such as the value of k (the number of nearest neighbors) and the distance metric (Euclidean distance). FA is inspired by the behavior of fireflies and is an efficient metaheuristic algorithm for solving optimization problems (Singh *et al.*, 2024). The mathematical model of the operation of FA is given below:

If attractiveness β is the distance between fireflies increases. The attractiveness of a firefly is determined using Equation 10.

$$\beta(r) = \beta_0 e^{-\gamma r^2} \quad (10)$$

Where:

- β_0 is the base attractiveness (at r=0r = 0r=0).
- γ is the light absorption coefficient, which controls the decrease in attractiveness with distance.
- γr is the distance between the two fireflies.

The distance r_{ij} between two fireflies i and j at positions $x_i$and x_j, respectively, is usually measured using Euclidean distance in a multidimensional search space as given in Equation 11.

$$r_{ij} = \sqrt{\sum_{k=1}^{d} \left(x_{i,j} - x_{j,k}\right)^2} \tag{11}$$

Where:

- $x_{i,j}$ and $x_{j,k}$ are the $k - th$components of the position vectors of fireflies i and j in a d-dimensional space.
- A firefly i is attracted to another firefly j if firefly j is brighter (if it has a better fitness value). The position of firefly i is updated using Equation 12:

$$x_i^{t+1} = x_i^t + \beta(r_{ij})(x_j^t - x_i^t) + \alpha\varepsilon_i^t \tag{12}$$

Where:

- x_i^t is the position of firefly iii at iteration ttt.
- $\beta(r_{ij})$ is the attractiveness of firefly j to firefly i, calculated using the attractiveness function.
- x_j^t is the position of the brighter firefly j at iteration t.
- α is the randomization parameter that controls the influence of random movement.
- ε_i^t is a random vector drawn from a uniform or Gaussian distribution to simulate randomness.
- The brightness I of a firefly is proportional to the value of the objective (fitness) function $f(x)$, which measures the quality of the solution represented by the firefly as shown in Equation 13.

$$I_i = f(x_i) \tag{13}$$

Where:

- I_i is the brightness (fitness) of firefly i.
- $f(x_i)$is the objective function evaluated at the position x_i.
- In this case, the fitness function is related to the accuracy and memory efficiency of the KNN algorithm, so:

$$f(x) = \frac{Accuracy}{Memory\ used} \tag{14}$$

The random movement component is controlled by α, which is multiplied by a random vector ε_i^t that adds randomness to the firefly's movement. The term $\varepsilon_i^t$is typically drawn from a uniform distribution $(0,1)$ or Gaussian distribution $N(0,1)$, adding exploration ability to the algorithm.

$\varepsilon_i^t \sim U(0,1)\ or\ N(0,1)$

This term ensures that the firefly has a small amount of random motion, which helps in exploring the search space and avoiding local optima.

The firefly algorithm iterates until one of the following conditions is met:

- A maximum number of iterations is reached.
- The change in fitness between successive iterations is less than a predefined threshold, indicating convergence.

Hence,

Movement of firefly i towards a brighter firefly j is governed by Equation 15:

$$x_i^{t+1} = x_i^t + \beta_0 e^{\gamma r_{ij}^2}\left(x_j^t - x_i^t\right) + \infty\varepsilon_i^t \tag{15}$$

Where r_{ij} is the Euclidean distance between firefly i and j, and ε_i^tintroduces randomness in the movement.

Equation 15 forms the core of the firefly optimization process, adjusting the positions of fireflies to converge towards the optimal solution based on their fitness values. The population of fireflies was initialized with random parameter settings for KNN (different values of k, distance metrics and so on) as shown in Algorithm 2.

Algorithm 2: Algorithm for Firefly Optimization

```
Begin
Initialize firefly population p randomly.
Initialize algorithm parameters $\alpha_0, \beta_0$ , and $\gamma$.
Determine fitness value using the objective function $f(x)$ for each firefly.
WHILE $(m < maxgeneration)$
        FOR $i = 1{:}p$
          FOR $j = 1{:}i$
              IF $(f(x_i) < f(x_j))$
              Move firefly i towards j using
    Calculate fitness value $\alpha$ again of all fireflies
              end if
      end for
        end for
end while
Find the current best firefly and choose it as representative of that class.
For each test tuple:
 Calculate the distance of the test tuple from each of the class representatives.
Assign that class to the test tuple from whose representative it has the least distance.
End.
```

The memory efficiency of KNN serves as the fitness function. The goal is to minimize memory consumption. Fireflies move based on their brightness, which corresponds to the fitness value. Brighter fireflies (those with better memory usage) will attract other fireflies, leading to parameter tuning. The algorithm iterates, adjusting the KNN parameters in each step, until an optimal or near-optimal solution is found. FA is used to tune the parameters of KNN to achieve improved memory efficiency. The results of this work show that the optimized KNN is more computationally efficient than the standard version, especially when dealing with large datasets. In this work, the performance of the optimized KNN-GLCM-FA approach was compared to a standard KNN implementation (without feature extraction and optimization).

Algorithm 3: Optimization of Memory-Efficient of KNN Using GLCM and Firefly Optimization

```
Input: Image dataset, number of nearest neighbors (K), firefly algorithm parameters
Output: Optimized KNN model with reduced memory usage and improved performance
Step 1. Data Collection and Preprocessing
    • Collection of the image dataset (Mammographic Images).
    • Normalization of the image data to ensure uniform pixel intensity across images.
    • Partitioning of the dataset into training and testing sets using 5-fold cross-validation.
Step 2. Feature Extraction using GLCM
    • Initialization of GLCM feature extraction parameters:
    • Set distance d = 1, 2, 4 pixels.
    • Set directions: horizontal (0°), vertical (90°), diagonal (45°), and anti-diagonal (135°).
 For each image in the dataset:
    • Compute the GLCM for each direction and distance.
    • Extract texture features: Contrast, Energy, Homogeneity, Correlation.
    • Store the extracted features for each image as the feature vector.
Step 3: Perform firefly optimization:
    While iteration < T do:
        For each firefly i:
 Evaluate the fitness of firefly i using the fitness function.
            For each firefly j ≠ i:
                If firefly j is brighter (has a higher fitness) than firefly i:
            Move firefly i towards firefly j based on the attraction rule:
Update position of firefly i = $K_i^{t+1} = K_i^t + \beta_0 e^{-\gamma r_{ij}^2}(K_j^t - K_i^t) + \alpha\epsilon_i^t$
Update the best firefly position (best KNN parameters) after each iteration.
    End while
Return the optimal KNN parameters (k, distance metric) obtained from the firefly algorithm.
Step 4: Compare the optimized GLCM-FA-KNN model with:
    • Baseline KNN (without GLCM and FA).
Step 5: Highlight improvements in memory usage.
End.
```

Formation of GLCM-FA-KNN

In a Memory-Efficient KNN model, the goal is to reduce the memory usage of the KNN algorithm while maintaining or improving its accuracy. The Gray Level Co-Occurrence Matrix (GLCM) is used to extract texture features from image data, and the Firefly Optimization Algorithm is applied to optimize KNN's parameters for better memory and computational efficiency as presented in Figure 3. The following outlines the steps and equations for the model, incorporating both GLCM and Firefly Optimization.

Key components:

- K-Nearest Neighbors (KNN): A non-parametric algorithm that classifies data points based on the majority vote of their nearest neighbors.
- Gray Level Co-Occurrence Matrix (GLCM): Extracts features from the image data, reducing the dimensionality and memory required for KNN.
- Firefly Algorithm: Optimizes KNN's parameters, such as the number of k-neighbors, to enhance memory efficiency while maintaining accuracy.
- Let $I(x, y)$ represent the intensity of the image at coordinates (x, y). The GLCM $p(i, j\backslash d, \theta)$ is used to extract texture features like contrast, correlation, homogeneity, energy, and entropy, reducing the feature space, Equation 16 shows the relationship among these parameters:
- $GLCM_{Features} = \{Contrast, Correlation, Homogeneity, Energy, Entropy\}$ (16)
- Instead of using the full image pixel data, which increases memory usage, only the five GLCM features were used as inputs to the KNN classifier. This significantly reduces memory consumption because the GLCM features compactly represent the texture information in the image. Using Equation 17,

The GLCM feature matrix FFF for an image is:

$$F = [f_i, f_2, f_3, \ldots\ldots\ldots.., f_m] \tag{17}$$

Where $f_i, f_2, f_3, \ldots\ldots\ldots.., f_m$ are the extracted GLCM features for the image, and mmm is the number of features (typically, $m = 5$).

The memory consumption M in a traditional KNN classifier is proportional to the size of the training dataset N and the feature dimensionality D and this is determined using Equation 18.

$$M_{KNN} = O(N \times D) \tag{18}$$

In the case of GLCM-based KNN, the dimensionality D is reduced from the original pixel count to the number of extracted GLCM features m as shown in Equation 19.

$$M_{KNN\_GLCM} = O(N * M) \tag{19}$$

Since $m \ll Dm$ the memory usage is significantly reduced, which improves the efficiency of KNN.

The FA is used to optimize the key parameters of the GLCM-KNN classifier, specifically the number of K-neighbors, as well as other parameters that affect memory and computational efficiency.

The objective function for the FA was designed to minimize memory usage while maximizing classification accuracy. The objective can be formulated as a weighted sum of memory and accuracy as shown in Equation 20.

$$Objective\ Function = w_i * \frac{1}{Accuracy} + w_2 * memory(K) \tag{20}$$

Where:

- $Accuracy(k)$is the classification accuracy of KNN with K-neighbors.
- Memory(k) is the memory usage of KNN, which depends on the number of K-neighbors and the dimensionality of the input features.
- w_i and w_2 are weights that balance the trade-off between accuracy and memory.
- FA Algorithm optimizes k based on the brightness (fitness) of each firefly, which represents a potential solution for the number of neighbors. The movement of a firefly i toward a brighter firefly j is given by Equation 21.

$$K_i^{t+1} = K_i^t + \beta_0 e^{-\gamma r_{ij}^2}\left(K_j^t - K_i^t\right) + \alpha\epsilon_i^t \tag{21}$$

Where:

- K_i^t is the number of neighbors for firefly iii at iteration t.
- $\beta_0$is the attractiveness constant.
- Υ is the light absorption coefficient.
- r_{ij} is the distance between fireflies i and j.
- $\alpha\epsilon_i^t$ is a randomization term to ensure diversity in the solutions.

The Firefly Algorithm iteratively updates k to find the optimal number of neighbors that minimizes the memory usage while maximizing classification accuracy.

The overall memory usage of the Memory-Efficient KNN with GLCM and Firefly Optimization is determined using Equation 22.

$$M_{KNN\_GLCM\_FA} = O(N * M * K^*) \tag{22}$$

Where:

- K^* is the optimized number of neighbors obtained through Firefly Optimization.
- N is the size of the training dataset.
- m is the number of GLCM features (typically small, e.g., $m = 5$).

Equation 22 reflects the optimized memory usage of the KNN algorithm after dimensionality reduction via GLCM and parameter tuning through Firefly Optimization.

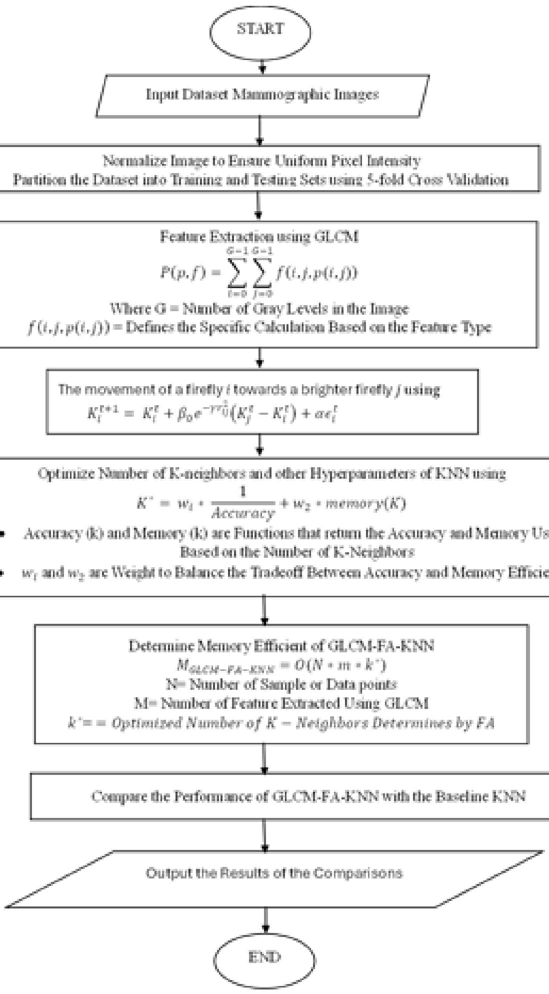
START
Input Dataset Mammographic Images
Normalize Image to Ensure Uniform Pixel Intensity
Partition the Dataset into Training and Testing Sets using 5-fold Cross Validation
Feature Extraction using GLCM
$P(p,f) = \sum_{i=0}^{G-1}\sum_{j=0}^{G-1} f(i,j,p(i,j))$
Where G = Number of Gray Levels in the Image
$f(i,j,p(i,j))$ = Defines the Specific Calculation Based on the Feature Type
The movement of a firefly i towards a brighter firefly j using
$K_i^{t+1} = K_i^t + \beta_0 e^{-\gamma r_{ij}^2}(K_j^t - K_i^t) + \alpha\epsilon_i^t$
Optimize Number of K-neighbors and other Hyperparameters of KNN using
$K^* = w_1 * \frac{1}{Accuracy} + w_2 * memory(K)$
Accuracy (k) and Memory (k) are Functions that return the Accuracy and Memory Usage Based on the Number of K-Neighbors
w_1 and w_2 are Weight to Balance the Tradeoff Between Accuracy and Memory Efficiency
Determine Memory Efficient of GLCM-FA-KNN
$M_{GLCM-FA-KNN} = O(N * m * k^*)$
N= Number of Sample or Data points
M= Number of Feature Extracted Using GLCM
$k^* = =$ Optimized Number of K − Neighbors Determines by FA
Compare the Performance of GLCM-FA-KNN with the Baseline KNN
Output the Results of the Comparisons
END

Figure 3: Flowchart for the optimization of memory-efficient of KNN Using GLCM and firefly optimization

3. RESULTS AND DISCUSSION

The study compares the memory efficiency of the GLCM-FA-KNN model against the traditional KNN algorithm over five folds of data partitioning, focusing on three data classes: Benign (B), Malignant (M), and Normal (N) selected samples and classified results are shown in Figure 4.

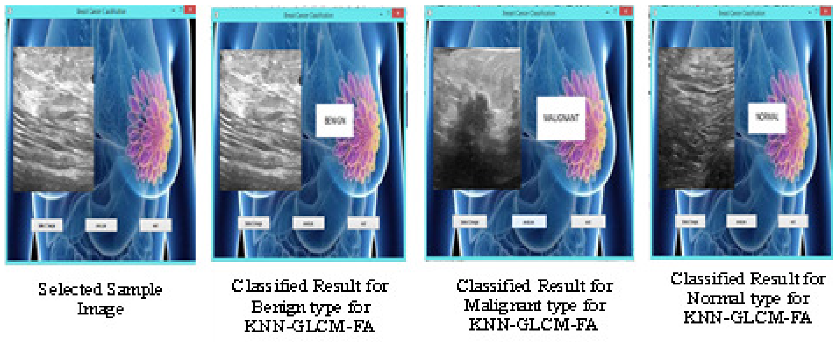
BENIGN
MALIGNANT
NORMAL
Selected Sample Image
Classified Result for Benign type for KNN-GLCM-FA
Classified Result for Malignant type for KNN-GLCM-FA
Classified Result for Normal type for KNN-GLCM-FA

Figure 4: Sample of the extracted features

The findings across these folds are consistent, revealing that GLCM-FA-KNN provides a significantly more memory-efficient solution. Below is an in-depth discussion of each fold, highlighting the improvements made by the GLCM-FA-KNN model. The result of the comparison of memory efficiency of GLCM-FA-KNN and traditional KNN for the first fold is shown in Table 1

Table 1: Comparison of GLCM-FA-KNN and KNN memory efficiency (KB) for first fold (KB)

Data class	GLCM-FA-KNN	KNN
B	5.32	9.99
M	5.31	10
N	5.3	9.99
Average	5.31	9.99

B: Benign M: Malignant N: Normal

It is observed from the table that GLCM-FA-KNN memory usage for B, M, and N data classes are 5.32 KB, 5.31 KB, and 5.30 KB, respectively, resulting in an average memory usage of 5.31 KB. While KNN memory usage for B, M, and N data classes are 9.99 KB, 10.00 KB, and 9.99 KB, respectively, with an average of 9.99 KB. From these results GLCM-FA-KNN shows a remarkable reduction in memory usage compared to the traditional KNN model, achieving a decrease of about 47% in the average memory requirement. The uniformity in memory usage across all data classes indicates that GLCM-FA-KNN is consistent and stable regardless of the classification type (Benign, Malignant, or Normal), showing robustness in memory efficiency, this is in line with (Tarakci and Ozkan, 2021). As shown in Table 2, the second fold of memory efficiency

Table 2: Comparison of GLCM-FA-KNN and KNN memory efficiency (KB) for second fold (KB)

Data class	GLCM-FA-KNN	KNN
B	5.3	9.88
M	5.31	9.9
N	5.31	9.72
Average	5.31	9.83

The memory usage remains low, with values of 5.30 KB (B), 5.31 KB (M), and 5.31 KB (N), leading to an average of 5.31 KB. Where KNN memory usage is recorded at 9.88 KB (B), 9.90 KB (M), and 9.72 KB (N), with an average of 9.83 KB. The memory efficiency improvement observed in the first fold is consistently maintained in this fold. KNN-Firefly's memory usage remains low, and the variance across the data classes is minimal, showcasing its effectiveness. The traditional KNN algorithm, while demonstrating some fluctuations in memory usage across the different classes, still averages a significantly higher value compared to KNN-Firefly. This consistency in efficiency reduction confirms the reliability of the Firefly optimization process applied to the KNN model, in line with (Jiang et al., 2022). Table 3 shows the results of FA-KNN compared with KNN from the third fold,

Table 3: Comparison of GLCM-FA-KNN and KNN memory efficiency (KB) for third fold (KB)

Data class	GLCM-FA-KNN	KNN
B	5.29	9.78
M	5.32	9.98
N	5.3	9.99
Average	5.3	9.91

KNN-Firefly memory values recorded are 5.29 KB (B), 5.32 KB (M), and 5.30 KB (N), resulting in an average of 5.30 KB compared to those of KNN that are 9.78 KB, 9.98 KB, and 9.99 KB respectively, with an average of 9.91 KB. Like previous folds, KNN-Firefly shows a significant reduction in memory usage, with a stable performance across all data classes. This consistency highlights the robustness of the GLCM and Firefly optimization approach, which efficiently minimizes the memory required for classification tasks. The traditional KNN model's memory usage is still nearly double that of the optimized model, emphasizing the scalability advantage of KNN-Firefly, especially in scenarios involving large datasets aligned with (Riquelme *et al.*, 2020). The results of the fourth fold is shown in Table 4, the same pattern is notices by generating

Table 4: Comparison of GLCM-FA-KNN and KNN memory efficiency (KB) for first fold (KB)

Data class	GLCM-FA-KNN	KNN
B	5.31	9.92
M	5.3	9.9
N	5.28	9.79
Average	5.3	9.87

KNN-Firefly memory usage of 5.31 KB (B), 5.30 KB (M), and 5.28 KB (N), with an average of 5.30 KB, while those of KNN are 9.92 KB, 9.90 KB, and 9.79 KB respectively, resulting in an average of 9.87 KB. KNN-Firefly's performance remains consistent, showing stable and reduced memory usage for each class, which averages to a significantly lower value compared to the standard KNN model. The pattern indicates that the Firefly Algorithm effectively optimizes the KNN parameters, achieving the dual goals of enhancing memory efficiency and ensuring that the algorithm remains performant across all types of data classes. This result highlights that KNN-Firefly is suitable for applications where memory resources are limited, such as mobile or embedded systems used for medical diagnostics, aligned with (Barrientos *et al.*, 2022).

It can be observed from Table 5 which shows the memory performance of KNN-Firefly from the fifth fold for B, M, and N classes as 5.30 KB, 5.30 KB, and 5.32 KB, respectively, with an average of 5.31 KB, while values recorded by KNN are 9.98 KB, 9.91 KB, and 9.78 KB, respectively, averaging 9.89 KB. The KNN-Firefly model consistently maintains a reduced memory profile, with minimal variation across different data classes. This stability across multiple folds suggests that the model's optimization process is both reliable and effective in managing memory usage regardless of data partitioning. The traditional KNN, while showing minor variations in memory usage, consistently requires a much larger memory allocation compared to KNN-Firefly, reinforcing the need for optimization strategies when deploying KNN in memory-constrained environments in accordance with (Ukey *et al.*, 2023b).

Table 5: Comparison of GLCM-FA-KNN and KNN memory efficiency (KB) for fifth fold

Data Class	GLCM-FA-KNN	KNN
B	5.3	9.98
M	5.3	9.91
N	5.32	9.78
Average	5.31	9.89

Overall Performance of KNN-Firefly is presented in Table 6. An overall average memory usage of 5.31 KB across the five folds, whereas the standard KNN model records an average memory usage of 9.90 KB as indicated in Table 6. This indicates that KNN-Firefly reduces memory usage by approximately 46.4% compared to the traditional KNN model as presented in Figure 5. The significant reduction highlights the effectiveness of combining GLCM for feature reduction and the Firefly Algorithm for parmeter optimization. The consistency of KNN-Firefly's performance across all folds shows its adaptability and reliability. By efficiently managing memory usage, KNN-Firefly is well-suited for large-scale applications, particularly in fields like medical imaging, where the processing and storage of large volumes of image data are common.

Table 6: Average memory usage efficiency of the five folds for GLCM-FA-KNN and KNN (KB)

Average	GLCM-FA-KNN	KNN
F1	5.31	9.99
F2	5.31	9.83
F3	5.3	9.91
F4	5.3	9.87
F5	5.31	9.89
Overall average	5.31	9.9

The ability of KNN-Firefly to maintain lower memory usage across all folds and data classes suggests that it is a scalable solution. This has important implications for real-world applications where computational resources are limited, such as mobile healthcare diagnostics, remote sensing, and automated quality control in manufacturing. The overall reduction in memory usage without compromising classification consistency or accuracy indicates that the combined approach of GLCM and Firefly Algorithm not only optimizes performance but also makes the KNN model more feasible for deployment in resource-constrained environments.

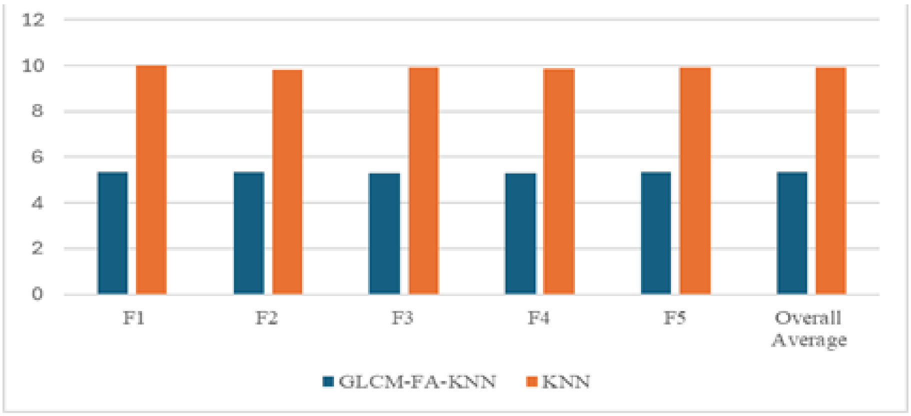

Figure 5: Comparison of the GLCM-FA-KNN and KNN memory requirement

4. CONCLUSION

This study enhanced the memory efficiency of the K-Nearest Neighbors (KNN) algorithm using the Gray Level Co-Occurrence Matrix (GLCM) for feature extraction and the Firefly Optimization Technique for further parameter optimization. The methodology was carefully designed to incorporate data collection and preprocessing, feature extraction through the GLCM-FA-KNN algorithm, and thorough evaluation and analysis of the results. The results across the five folds demonstrate a consistent and significant improvement in memory efficiency when comparing the optimized KNN-Firefly model with the traditional KNN approach. Overall, the average memory usage across all folds for KNN-Firefly was consistently 5.31 KB, while the standard KNN recorded an average of 9.90 KB. This reflects an approximate 46.4% reduction in memory usage, clearly illustrating the efficacy of the GLCM and Firefly Algorithm integration in optimizing KNN for memory efficiency. The study conclusively shows that the combined use of GLCM for feature extraction and the Firefly Algorithm for parameter tuning effectively enhances the memory efficiency of the KNN algorithm without compromising accuracy. This optimized approach is highly suitable for large-scale applications, particularly in memory-constrained environments such as medical imaging and mobile healthcare diagnostics. The consistent and stable performance of the KNN-Firefly model across multiple data partitions and classes further validates its robustness and adaptability, making it a valuable contribution to the field of computational efficiency in machine learning.

5. CONFLICT OF INTEREST

There is no conflict of interest associated with this work.

REFERENCES

Abdolrazzagh-Nezhad, M., Mahyabadi, S.P. and Ebrahimpoor, A., (2020). Improving KNN by Gases Brownian Motion Optimization Algorithm to Breast Cancer Detection. Data Science: Journal of Computing and Applied Informatics, 4(1), pp.1-15.

Albert, T., (2023). The Physiological Mechanisms of Triple Negative Breast Cancer in African American Women. Georgetown Medical Review, 6(1).

AlMutlaq, A.J., Jawawi, D.N. and Arbain, A.F.B., (2023). Weight optimization based on firefly algorithm for analogy-based effort estimation. International Journal of Advanced Computer Science and Applications, 14(6).

Barrientos, R.J., Riquelme, J.A., Hernández-García, R., Navarro, C.A. and Soto-Silva, W., 2022. Fast kNN query processing over a multi-node GPU environment. The Journal of Supercomputing, pp.1-27.

Barrientos, R.J., Riquelme, J.A., Hernández-García, R., Navarro, C.A. and Soto-Silva, W., 2022. Fast kNN query processing over a multi-node GPU environment. The Journal of Supercomputing, pp.1-27.

Benyahia, S., Meftah, B. and Lézoray, O., (2022). Multi-features extraction based on deep learning for skin lesion classification. Tissue and Cell, 74, p.101701

Djunaidi, K., Agtriadi, H.B., Kuswardani, D. and Purwanto, Y.S., (2021). Gray level co-occurrence matrix feature extraction and histogram in breast cancer classification with ultrasonographic imagery. Indonesian Journal of Electrical Engineering and Computer Science, 22(2), pp.795-800.

Hanker, A.B., Sudhan, D.R. and Arteaga, C.L., (2020). Overcoming endocrine resistance in breast cancer. Cancer cell, 37(4), pp.496-513.

Hussain, L., Alsolai, H., Hassine, S.B.H., Nour, M.K., Duhayyim, M.A., Hilal, A.M., Salama, A.S., Motwakel, A., Yaseen, I. and Rizwanullah, M., (2022). Lung cancer prediction using robust machine learning and image enhancement methods on extracted gray-level co-occurrence matrix features. Applied Sciences, 12(13), p.6517.

Jiang, T., Zhang, B., Lin, D., Gao, Y. and Li, Q., (2022). Efficient parallel processing of high-dimensional spatial k NN queries. Soft Computing, 26(22), pp.12291-12316.

Juna, A., Umer, M., Sadiq, S., Karamti, H., Eshmawi, A.A., Mohamed, A. and Ashraf, I., (2022). Water quality prediction using KNN imputer and multilayer perceptron. Water 2022, 14, 2592.

Kaliappan, J., Bagepalli, A.R., Almal, S., Mishra, R., Hu, Y.C. and Srinivasan, K., (2023). Impact of Cross-validation on Machine Learning models for early detection of intrauterine fetal demise. Diagnostics, 13(10), p.1692.

Khan, I.U., Aslam, N., Alshehri, R., Alzahrani, S., Alghamdi, M., Almalki, A. and Balabeed, M., (2021). Cervical cancer diagnosis model using extreme gradient boosting and bioinspired firefly optimization. Scientific programming, 2021(1), p.5540024.

Kherif, O., Benmahamed, Y., Teguar, M., Boubakeur, A. and Ghoneim, S.S., (2021). Accuracy improvement of power transformer faults diagnostic using KNN classifier with decision tree principle. IEEE Access, 9, pp.81693-81701.

Koosha, H.R., Ghorbani, Z. and Nikfetrat, R., (2022). A clustering-classification recommender system based on firefly algorithm. Journal of AI and Data Mining, 10(1), pp.103-116.

Krishna, S. and George, B., 2021. An affordable solution for the recognition of abnormality in breast thermogram. Multimedia Tools and Applications, 80(18), pp.28303-28328.

Labrada, A. and Barkana, B.D., (2022)., July. Breast cancer diagnosis from histopathology images using supervised algorithms. In 2022 IEEE 35th International Symposium on Computer-Based Medical Systems (CBMS) (pp. 102-107).

Lahmiri, S., (2023). Integrating convolutional neural networks, kNN, and Bayesian optimization for efficient diagnosis of Alzheimer's disease in magnetic resonance images. Biomedical Signal Processing and Control, 80, p.104375.

Lopez, K.M.M. and Magboo, M.S.A., (2020). A clinical decision support tool to detect invasive ductal carcinoma in histopathological images using support vector machines, Naïve-Bayes, and K-nearest neighbor classifiers. In Machine Learning and Artificial Intelligence (pp. 46-53).

Park, M., Kim, D., Ko, S., Kim, A., Mo, K. and Yoon, H., (2022). Breast cancer metastasis: mechanisms and therapeutic implications. International journal of molecular sciences, 23(12), p.6806.

Peng, L., Huang, L., Su, Q., Tian, G., Chen, M. and Han, G., (2024). LDA-VGHB: identifying potential lncRNA–disease associations with singular value decomposition, variational graph auto-encoder and heterogeneous Newton boosting machine. Briefings in Bioinformatics, 25(1), p.bbad466.

Rahman, M.A., Hossain, M.F., Hossain, M. and Ahmmed, R., (2020). Employing PCA and t-statistical approach for feature extraction and classification of emotion from multichannel EEG signal. Egyptian Informatics Journal, 21(1), pp.23-35. S1110866519301720

Riesaputri, D.F., Sari, C.A., De Rosal, I.M.S. and Rachmawanto, E.H., (2020)., September. Classification of breast cancer using PNN classifier based on GLCM feature extraction and GMM segmentation. In 2020 International Seminar on Application for Technology of Information and Communication (iSemantic) (pp. 83-87.

Riquelme, J.A., Barrientos, R.J., Hernández-García, R. and Navarro, C.A., (2020)., November. An exhaustive algorithm based on GPU to process a kNN query. In 2020 39th International Conference of the Chilean Computer Science Society (SCCC) (pp. 1-8).

Sabry, F., (2023). K Nearest Neighbor Algorithm: Fundamentals and Applications (Vol. 28). One Billion Knowledgeable.

Sinaga, D., Agustina, F., Setiyanto, N.A., Suprayogi, S. and Jatmoko, C., (2021). Classification of bird based on face types using gray level co-occurrence matrix (GLCM) feature extraction based on the k-nearest neighbor (K-NN) algorithm. Journal of Applied Intelligent System, 6(2), pp.111-119.

Singh, R.M., Sikka, G. and Awasthi, L.K., (2024). A Modified Levy Flight Firefly-Based Approach to Optimize Turnaround Time in Fog Computing Environments. IETE Journal of Research, pp.1-11.

Tarakci, F. and Ozkan, I.A., (2021). Comparison of classification performance of kNN and WKNN algorithms. https://acikerisim.selcuk.edu.tr/items/338bcf79-e3a0-40d2-8968-4928a8cd4a39

Tiwari, V.R. (2023)., Developments in KD Tree and KNN Searches. International Journal of Computer Applications, 975 185 (17), p.17-23.8887.

Ukey, N., Yang, Z., Li, B., Zhang, G., Hu, Y. and Zhang, W., (2023). Survey on exact knn queries over high-dimensional data space. Sensors, 23(2), p.629.

Ukey, N., Yang, Z., Li, B., Zhang, G., Hu, Y. and Zhang, W., (2023). Survey on exact knn queries over high-dimensional data space. Sensors, 23(2), p.629.

Yadav, S.S. and Jadhav, S.M., (2022). Thermal infrared imaging based breast cancer diagnosis using machine learning techniques. Multimedia Tools and Applications, pp.1-19.

Zahirah, D., Purnawansyah, P., Kurniati, N. and Darwis, H., (2024). Digital Image Classification of Herbal Leaves using KNN and CNN With GLCM Features. Jurnal Teknik Informatika (JUTIF), 5(1), pp.61-67.

Zhao, F., Zhang, M., Zhou, S. and Lou, Q., (2024). Detection of network security traffic anomalies based on machine learning KNN method. Journal of Artificial Intelligence General science (JAIGS) ISSN: 3006-4023, 1(1), pp.209-218.

Zhu, Y., Yu, X., Chandraker, M. and Wang, Y.X., (2020). Private-knn: Practical differential privacy for computer vision. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition pp. 11854-11862.