

# Feature Extraction for Image Analysis using Genetic Algorithm

MAULANA ABUL KALAM AZAD  
UNIVERSITY OF TECHNOLOGY,  
WEST BENGAL



Maulana Abul Kalam Azad University of Technology, West Bengal.

A thesis in partial fulfilment of the requirement for the degree of Bachelor of Computer Science

& Engineering

July – 2025

*By*

**Name     Subhradip Sinha**

**Roll No.   10000122057**

**Reg No.   221000112121**

*Under the Guidance of*

**Dr. Sriyankar Acharyya**

*Professor,*

*Department of Computer Science & Engineering Maulana Abul Kalam  
Azad University of Technology Kolkata – 742449, W.B., India Faculty of  
Maulana Abul Kalam Azad University of Technology*

## **CERTIFICATE**

This is to certify that the dissertation entitled “**Feature Extraction for Image Analysis using Genetic Algorithm**” has been carried out by **Subhradip Sinha** (University Roll No.: **10000122057**, Registration No.: **221000112121** of **2022-23**) under my guidance and supervision and is hereby accepted in partial fulfilment of the requirements for the degree of **Bachelor in Computer Science & Engineering**.

The research results presented in this thesis have not been included in any other paper submitted for the award of any degree to any other University or Institute.

-----  
Dr. Sriyankar Acharyya  
Professor  
Dept. of Computer Science & Engineering

-----  
Head of the Department  
Dr. Saikat Basu  
HoD and Associate Professor  
Dept. of Computer Science & Engineering

## **CERTIFICATE OF ORIGINALITY**

I certify that:

- I. The work presented in this report is original and has been completed by me under the supervision of my advisor(s).
- II. This work has not been submitted to any other institution for any degree or diploma.
- III. I have adhered to the guidelines provided by the Institute while preparing this report.
- IV. I have followed the norms and guidelines outlined in the Institute's Ethical Code of Conduct.
- V. Whenever I have used materials (data, theoretical analysis, figures, and text) from other sources, I have properly credited them by citing in the report and including their details in the references. Additionally, I have obtained permission from the copyright owners where necessary.

-----  
Signature of the student

## CERTIFICATE OF APPROVAL

MAULANA ABUL KALAM AZAD  
UNIVERSITY OF TECHNOLOGY,  
WEST BENGAL



This is to certify that the Dissertation Report titled “**Feature Extraction for Image Analysis using Genetic Algorithm**” submitted by **Subhradip Sinha** to Maulana Abul Kalam Azad University of Technology (MAKAUT), West Bengal, India, is a record of genuine project work carried out by him under my/our supervision and guidance. It is worthy of consideration for the award of the degree of **Bachelor of Technology in Computer Science & Engineering**.

-----  
Signature of the Project Supervisor  
Dr. Sriyankar Acharyya  
Professor  
Dept. of CSE, MAKAUT, WB

-----  
Head of the Department  
Dr. Saikat Basu  
HoD and Associate Professor  
Dept. of Computer Science & Engineering

## ACKNOWLEDGEMENTS

I would like to express our sincere gratitude to everyone who has contributed to the successful completion of our project, "Feature Extraction for Image Analysis using Genetic Algorithm"

First and foremost, i would like to thank our project supervisor, Dr. Sriyankar Acharyya, for his invaluable guidance, support, and encouragement throughout the course of this project. his insights and expertise were crucial in shaping our work and overcoming various challenges.

I also grateful to the faculty and staff of the Computer Science and Engineering Department at MAKAUT, WB, for providing us with the necessary resources and a conducive environment for our research and development.

A Special thanks to our friends and colleagues for their continuous support and for providing a collaborative atmosphere that motivated us to work diligently.

I would also like to extend our appreciation to our families for their unwavering support and understanding throughout this journey. Their encouragement and belief in our abilities have been a constant source of inspiration.

Lastly, i would like to thank all the individuals who directly or indirectly contributed to this project. Your assistance and contributions have been invaluable to the successful completion of our work.

Thank you all.

---

Subhradip Sinha

# CONTENTS

<b>Title</b>	<b>Page No.</b>
<b>Certificate</b>	<b>I</b>
<b>Certificate of Originality</b>	<b>II</b>
<b>Certificate of Approval</b>	<b>III</b>
<b>Acknowledgements</b>	<b>IV</b>
<b>Abstract</b>	<b>6</b>
<b>1.Introduction</b>	<b>7</b>
<b>2.Problem Description</b>	<b>8</b>
<b>3.Literature Survey</b>	<b>9 - 11</b>
<b>4.Methodology</b>	<b>12</b>
4.1Problem Formulation	<b>12</b>
4.2Genetic Algorithm components	<b>12 - 14</b>
4.3Integration with Image Analysis Pipeline	<b>15</b>
4.4Dataset and Evaluation metrics	<b>16</b>
4.5Mathematical Calculation	<b>16 - 18</b>
<b>5.Future Plan</b>	<b>20</b>
<b>6.References</b>	<b>21</b>

## **Abstract**

Image analysis is a critical field with applications ranging from medical diagnostics to autonomous navigation. A fundamental step in image analysis is feature extraction, which involves transforming raw image data into a set of meaningful and representative features. Traditional feature extraction methods often struggle with high dimensionality, redundancy, and the challenge of identifying optimal feature subsets for specific tasks. This report proposes a novel approach to feature extraction using Genetic Algorithms (GAs). GAs are powerful metaheuristic optimization techniques inspired by natural selection, capable of exploring large search spaces efficiently. This project aims to leverage GAs to automatically select and optimize feature subsets from images, thereby enhancing the accuracy, robustness, and computational efficiency of subsequent image analysis tasks such as classification or segmentation. The methodology outlines the formulation of feature extraction as an optimization problem, defining chromosome encoding, fitness functions based on classification performance, and the application of standard GA operators. Expected outcomes include improved performance metrics and reduced computational overhead compared to conventional methods.

# 1.Introduction

The rapid proliferation of digital imaging technologies has made image analysis an indispensable tool across numerous scientific, industrial, and everyday applications. From diagnosing diseases through medical scans to enabling facial recognition in security systems, the ability to extract meaningful information from visual data is paramount. At the core of effective image analysis lies **feature extraction**, a process that transforms raw pixel data into a compact and informative representation. These "features" are quantitative measures that characterize different aspects of an image, such as edges, textures, shapes, or color distributions, making them amenable to computational processing.

The quality of extracted features directly impacts the performance of subsequent image analysis tasks, including classification, segmentation, object detection, and recognition. Poorly chosen or redundant features can lead to high computational costs, reduced accuracy, and difficulty in generalizing to new data. Traditional feature extraction methods, while effective in certain contexts (e.g., SIFT, HOG, PCA), often require domain expertise, are computationally intensive for high-dimensional data, or may not yield the most optimal feature set for a specific problem.

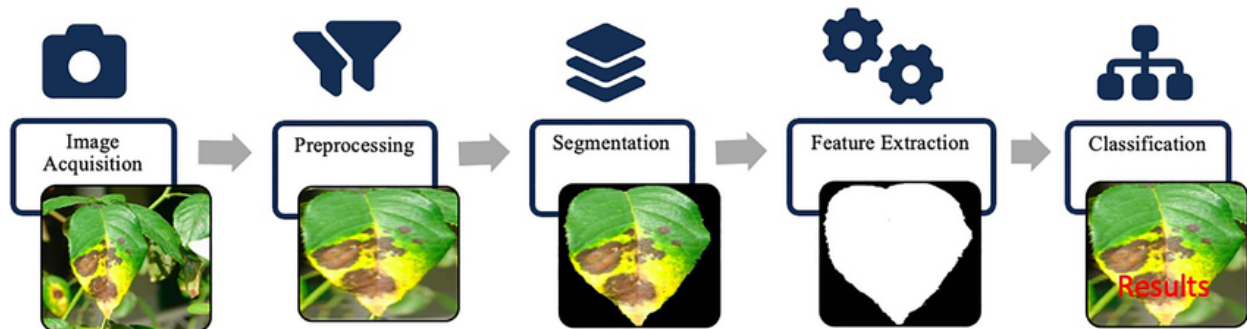
In recent years, metaheuristic optimization algorithms have gained significant attention for their ability to solve complex problems where traditional analytical methods fall short. Among these, **Genetic Algorithms (GAs)** stand out due to their robust search capabilities inspired by biological evolution. GAs operate on a population of potential solutions (chromosomes), iteratively improving them through processes analogous to natural selection, crossover, and mutation. This evolutionary approach allows GAs to explore vast solution spaces and converge towards optimal or near-optimal solutions without getting trapped in local optima.

This project proposes to harness the power of Genetic Algorithms to address the challenges in feature extraction for image analysis. By framing the feature extraction problem as an optimization task, GAs can be employed to automatically discover and select the most discriminative and non-redundant feature subsets from a larger pool of potential features. This approach promises to yield more efficient, accurate, and robust image analysis systems, reducing the need for manual feature engineering and enhancing adaptability across diverse imaging applications.



## 2.Problem Description

Traditional image analysis methods often rely on feature extraction techniques that result in **high-dimensional and redundant feature sets**, leading to the "**curse of dimensionality**," increased **computational cost**, and **sub-optimal performance** due to features not being specifically tailored to the task at hand. This project aims to address these limitations by developing a novel framework that applies **Genetic Algorithms (GAs)** to **automatically select optimal feature subsets** for various image analysis tasks, thereby reducing dimensionality, removing redundancy, and identifying the most discriminative features. Our objectives include designing a robust chromosome encoding scheme and a task-specific fitness function, implementing and evaluating the GA-based method on diverse image datasets, and quantitatively comparing its performance (accuracy, computational efficiency, feature set size) against traditional approaches to demonstrate GAs' potential in **automating and optimizing the feature engineering process**.



### 3.Literature Survey

Feature extraction is a well-researched area in computer vision and image processing. Over the decades, numerous techniques have been developed, broadly categorized into handcrafted features and learned features. This section reviews relevant literature on traditional feature extraction methods and the emerging role of Genetic Algorithms in image analysis.

#### 3.1 Traditional Feature Extraction Methods

Traditional methods rely on mathematical models and domain knowledge to extract predefined characteristics from images.

- Edge Detection: Operators like Sobel, Prewitt, Canny, and Laplacian of Gaussian (LoG) identify boundaries of objects. Canny is particularly popular for its multi-stage approach, including non-maximum suppression and hysteresis thresholding, providing robust edges.
- Corner Detection: Harris Corner Detector and Shi-Tomasi Corner Detector identify points of high curvature, which are stable under rotation and illumination changes.
- Texture Descriptors:
  - Gray-Level Co-occurrence Matrix (GLCM): Quantifies texture by calculating how often different gray-level pixel pairs occur in an image. Features like contrast, correlation, energy, and homogeneity are derived from GLCM.
  - Local Binary Patterns (LBP): Describes the local texture by thresholding the neighborhood of each pixel with the center pixel's value. LBP is computationally efficient and robust to monotonic illumination changes.
- Shape Descriptors:
  - Hu Moments: Seven invariant moments derived from central moments, invariant to translation, scale, and rotation. Useful for shape recognition.
  - Fourier Descriptors: Represent a shape's boundary in the frequency domain, providing a compact and invariant representation.
- Scale-Invariant Feature Transform (SIFT): Developed by Lowe (2004), SIFT features are robust to changes in scale, rotation, illumination, and minor perspective changes. It detects keypoints and describes them using local gradient orientations. SIFT has been widely used in object recognition and image matching.
- Speeded Up Robust Features (SURF): Bay et al. (2006) proposed SURF as a faster alternative to SIFT, using integral images for rapid computation of Hessian matrix approximations.
- Histogram of Oriented Gradients (HOG): Dalal and Triggs (2005) introduced HOG for object detection, particularly human detection. It describes local object appearance and shape by the distribution of intensity gradients or edge directions.
- Principal Component Analysis (PCA): A dimensionality reduction technique that transforms data into a new coordinate system such such that the greatest variance by any projection lies

on the first coordinate (called the first principal component), the second greatest variance on the second coordinate, and so on. While not strictly a feature *extraction* method in the traditional sense, PCA can be used to derive a new set of features from existing ones.

While these methods have proven effective, they often generate a large number of features, leading to redundancy and increased computational burden. The challenge lies in selecting the most informative subset of these features for a given task.

### 3.2 Genetic Algorithms in Image Analysis

Genetic Algorithms (GAs) are a class of evolutionary algorithms inspired by the process of natural selection. They are particularly well-suited for optimization problems where the search space is large and complex. GAs operate on a population of candidate solutions, evolving them over generations using genetic operators like selection, crossover, and mutation.

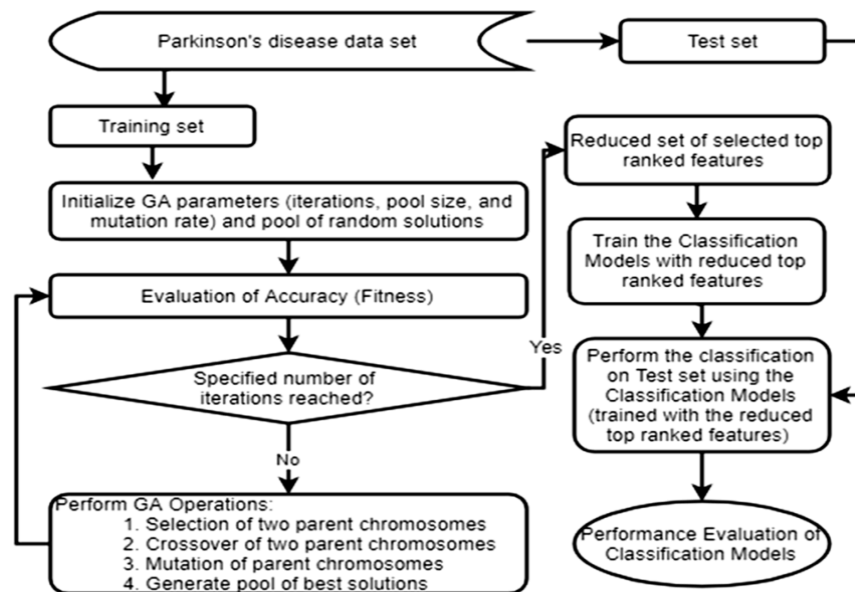
The application of GAs in image processing and computer vision has been diverse:

- **Image Segmentation:** GAs have been used to optimize parameters for image segmentation algorithms (e.g., thresholding, clustering) or to directly evolve segmentation masks. For instance, Bhandari et al. (2018) used a GA to optimize parameters for K-means clustering in image segmentation.
- **Image Registration:** GAs can optimize transformation parameters (translation, rotation, scaling) to align two or more images.
- **Image Restoration and Enhancement:** GAs have been applied to optimize filters for noise reduction or contrast enhancement.
- **Object Recognition:** GAs can be used to optimize classifier parameters or to select optimal features for object recognition tasks.

More specifically, GAs have shown promise in feature selection for various domains, including image analysis:

- **Feature Subset Selection:** The core idea is to treat the selection of features as an optimization problem where the GA searches for the best subset of features that maximizes a predefined performance metric (e.g., classification accuracy) while potentially minimizing the number of features.
  - P. N. Tan et al. (2005) discuss feature subset selection as a critical step in data mining, and GAs are often cited as effective search strategies for this problem.
  - For image data, researchers have combined GAs with various classifiers (e.g., SVM, k-NN, Neural Networks) to evaluate the fitness of a feature subset. For example, Al-Ani and Al-Dahoud (2009) used a GA for feature selection in facial recognition, demonstrating improved accuracy.
  - In medical image analysis, GAs have been employed to select relevant features for disease diagnosis. For instance, a study by Zhang et al. (2010) applied GA for feature selection in MRI brain image classification.

The existing literature highlights the potential of GAs to automate and optimize the feature selection process. However, there is still scope for exploring more sophisticated GA implementations, designing more robust fitness functions tailored to specific image analysis challenges, and integrating GAs with a wider range of initial feature extraction techniques to create a comprehensive and adaptive feature engineering pipeline. This project aims to contribute to this area by providing a detailed methodology for GA-based feature extraction and demonstrating its efficacy.



## 4. Methodology

This section details the proposed methodology for implementing feature extraction using a Genetic Algorithm. The approach involves framing the feature extraction problem as an optimization task, defining the GA components, and integrating them into a complete image analysis pipeline.

### 4.1 Problem Formulation

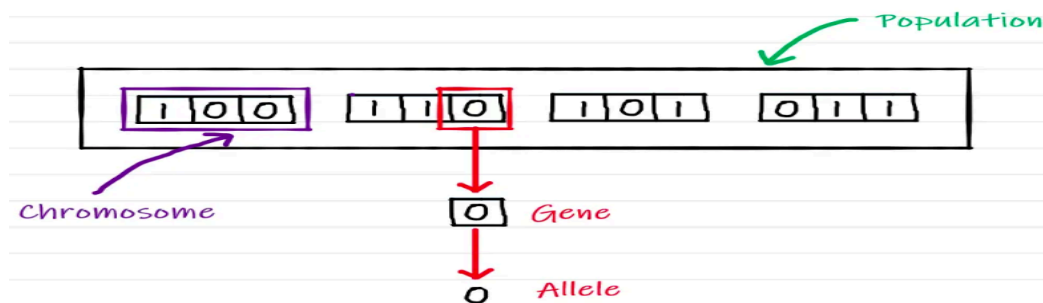
The goal is to find an optimal subset of features that maximizes the performance of a subsequent image analysis task (e.g., classification accuracy) while potentially minimizing the number of features. This can be formulated as a search problem in a high-dimensional space of all possible feature subsets. Given a set of  $N$  initial features, there are  $2^N$  possible subsets, making exhaustive search computationally infeasible for large  $N$ . Genetic Algorithms are well-suited for exploring this vast search space.

### 4.2 Genetic Algorithm Components

To achieve this optimization, we rely on the core components of a Genetic Algorithm, each playing a vital role in evolving better solutions over successive generations. Let's break down how each piece fits into our feature extraction puzzle.

#### 4.2.1 Chromosome Encoding (Representation)

First up, we need a way to represent our potential solutions – these are called 'chromosomes' in GA terms. For our feature selection problem, a chromosome simply tells us which features are 'in' and which are 'out'. We'll use a straightforward binary encoding for this. Think of each chromosome as a string of ones and zeros, with its length matching the total number of initial features we started with. If a '1' appears at a certain position, it means we've chosen that particular feature; a '0' means we've left it out. For instance, if we began with five features (F1, F2, F3, F4, F5), a chromosome like 10110 would indicate that features F1, F3, and F4 are part of this specific subset.



### 4.2.2 Fitness Function

The fitness function evaluates the quality of each chromosome (feature subset). This function gives each chromosome a 'score' based on how well its selected features perform. A higher score means a better, more effective feature set. For classification tasks, our fitness function will primarily focus on the classification accuracy achieved by a model using *only* those features. We'll train a classifier (like a Support Vector Machine (SVM) or a K-Nearest Neighbors (k-NN) model) with the features specified by the chromosome and then measure its accuracy on a separate validation dataset. But we don't just want accuracy; we also want to keep our feature set lean! So, to encourage simpler, less redundant solutions, we'll add a small penalty if a chromosome selects too many features. This helps us find the perfect spot: high accuracy with a manageable number of features. Here's how we'll calculate it:

$$\text{Fitness}(C) = \alpha \times \text{Accuracy}(C) - \beta \times \text{Size}(C)$$

Where:

- $C$  is the current chromosome (feature subset).
- $\text{Accuracy}(C)$  is the classification accuracy achieved using features in  $C$ .
- $\text{Size}(C)$  is the number of features in  $C$ .
- $\alpha$  and  $\beta$  are weighting coefficients, typically  $\alpha \gg \beta$  to prioritize accuracy while still penalizing large sizes.

Let's consider an example for the fitness function. Suppose our classifier achieves an accuracy of 0.85 (85%) with a subset of 10 features. If we set  $\alpha=100$  (to emphasize accuracy) and  $\beta=1$  (to penalize size), the fitness would be  $100 \times 0.85 - 1 \times 10 = 85 - 10 = 75$ . Now, imagine another subset with 0.84 accuracy but only 5 features. Its fitness would be  $100 \times 0.84 - 1 \times 5 = 84 - 5 = 79$ . In this case, even with slightly lower accuracy, the smaller feature set might be preferred due to its higher fitness score, demonstrating the trade-off. Tuning  $\alpha$  and  $\beta$  is crucial to balance these two objectives.

### 4.2.3 Population Initialization

An initial population of chromosomes is generated randomly. Each chromosome is a binary string, with each bit randomly set to 0 or 1. The population size is a critical parameter, influencing the diversity and convergence speed of the GA.

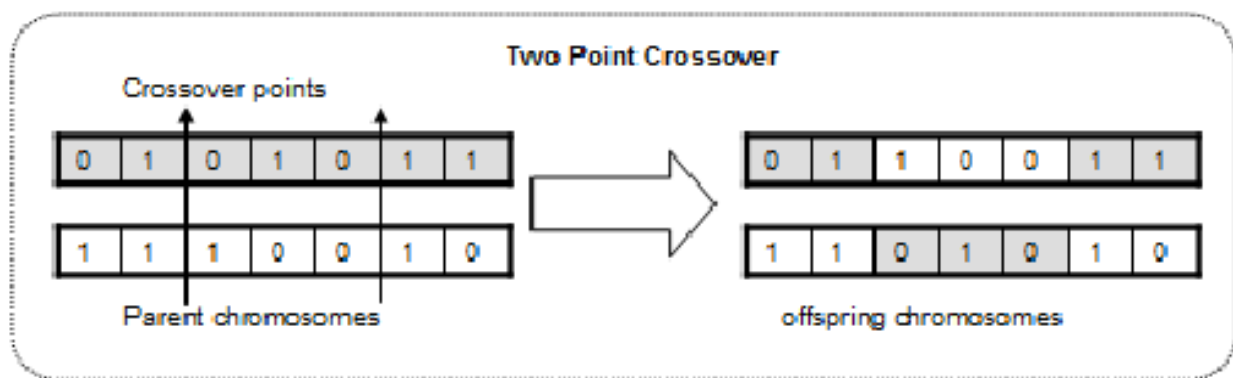
### 4.2.4 Selection

Once we have a population, the 'selection' phase begins. This is where we pick the 'fittest' chromosomes – those with the highest scores from our fitness function – to be the 'parents' for the next generation. The idea is simple: better solutions have a higher chance of passing on their 'genes'. We might use techniques like 'Roulette Wheel Selection,' where each chromosome gets a slice of a metaphorical wheel proportionate to its fitness, and we spin it to pick parents. Or, 'Tournament Selection' could be used, where we randomly pick a few chromosomes and simply

choose the best one among them. This helps ensure that promising feature combinations continue to evolve.

#### 4.2.5 Crossover (Recombination)

Crossover combines genetic material from two parent chromosomes to create new offspring chromosomes, promoting the exploration of the search space. This process can be implemented through various methods: One-Point Crossover involves selecting a single random point and swapping the genetic material after this point between the two parents; Two-Point Crossover extends this by choosing two random points and swapping the segment of genetic material located between them; and Uniform Crossover, which operates at a finer grain, determines each bit in the offspring chromosome by inheriting it from either parent with a predefined probability, typically 0.5.



#### 4.2.6 Mutation

In a Genetic Algorithm, **mutation** introduces random changes to chromosomes (like flipping a bit in binary encoding) to prevent stagnation in local optima and maintain genetic diversity, allowing for broader exploration of the solution space.

#### 4.2.7 Termination Criteria

The Genetic Algorithm (GA) stops when it meets certain conditions. These "termination criteria" signal the end of its evolution. It might stop after a set number of **generations** (which are like rounds of the algorithm), or when the **best solution** it has found reaches a certain level of "fitness" (meaning it's good enough). Another reason it might stop is if the best solution doesn't get noticeably better for a while. Once any of these conditions are met, the GA will show us the best set of features it discovered.



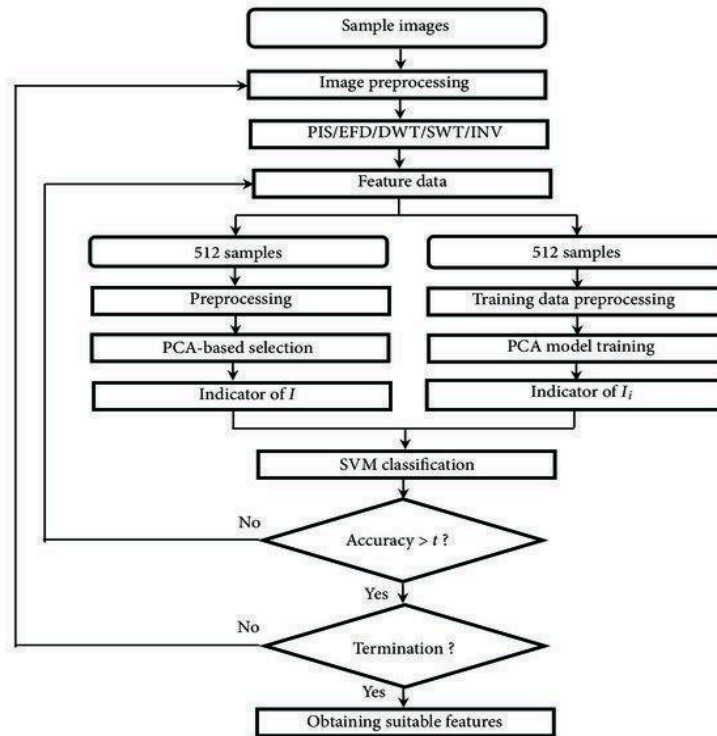
### 4.3 Integration with Image Analysis Pipeline

First, we begin with **raw images**. We might do some **preprocessing** to clean them up, like reducing noise or resizing, to get them ready. Next, we generate a large pool of **initial features** from these images using standard techniques such as HOG or LBP. The more features we have at this stage, the better, as the GA will later select the most effective ones.

Before running the GA, we **split our dataset** into three parts: a training set for teaching our classifier, a validation set for the GA to evaluate its choices, and a test set for a final, unbiased performance check.

Then comes the heart of the process: **Genetic Algorithm execution**. This is an iterative cycle where we start with a random group of potential solutions (chromosomes). In each **generation**, we assess how well each solution performs. We then select the best ones as "parents," combine them through **crossover** to create new solutions, and introduce small, random **mutations** to keep things diverse. This continues until our predefined stopping rules (termination criteria) are met.

Once the GA finishes, it identifies the **optimal feature subset**—the combination of features that performed best throughout the process. Finally, we use this optimal subset to **train our final model** on the combined training and validation data. We then evaluate this model on the completely unseen **test set** to see how well our GA-optimized feature extraction performs in a real-world setting.





## 4.4 Dataset and Evaluation Metrics

### 4.4.1 Dataset

The proposed methodology can be applied to various image datasets, such as:

Medical Images (e.g., MRI scans for tumor detection, X-rays for fracture identification), Object Recognition Datasets (e.g., CIFAR-10, MNIST, custom datasets for specific object categories), Facial Recognition Datasets: (e.g., Labeled Faces in the Wild (LFW)), Remote Sensing Images: (e.g., satellite images for land cover classification), The dataset should have a sufficient number of labeled images to allow for robust training, validation, and testing.

### 4.4.2 Evaluation Metrics

The performance of the GA-based feature extraction method will be evaluated using standard metrics for classification tasks:

- Accuracy:  $(\text{True Positives} + \text{True Negatives}) / \text{Total Samples}$
- Precision:  $\text{True Positives} / (\text{True Positives} + \text{False Positives})$
- Recall (Sensitivity):  $\text{True Positives} / (\text{True Positives} + \text{False Negatives})$
- F1-Score:  $2 \times (\text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall})$  (harmonic mean of precision and recall)
- Confusion Matrix: Provides a detailed breakdown of correct and incorrect classifications for each class.
- Feature Set Size: The number of features selected by the GA. This is a crucial metric for evaluating dimensionality reduction.
- Computational Cost: Time taken for the GA to converge and the time taken for classification using the selected features.

Comparison will be made against traditional methods using the full set of initial features and potentially other feature selection techniques (e.g., Recursive Feature Elimination, PCA).

## 4.5 Mathematical Formulations and Calculations

This section provides a more detailed look into the mathematical calculations involved, particularly for evaluating the fitness of chromosomes and the final model performance.

### 4.5.1 Confusion Matrix Components

For a binary classification problem (e.g., presence/absence of a feature, or classifying into two classes), the outcomes can be categorized as follows:

- True Positives (TP): Instances correctly predicted as positive.
- True Negatives (TN): Instances correctly not predicted as negative.
- False Positives (FP): Instances incorrectly predicted as positive (Type I error).
- False Negatives (FN): Instances incorrectly not predicted as negative (Type II error).

These components form the basis for calculating various performance metrics.

#### 4.5.2 Classification Performance Metrics

Given TP, TN, FP, and FN, the following metrics are calculated:

- Accuracy: Measures the proportion of correctly classified instances out of the total instances.

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

- Precision: Measures the proportion of correctly predicted positive instances among all instances predicted as positive. It answers: "Of all items the classifier labeled as positive, how many are actually positive?"

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

- Recall (Sensitivity or True Positive Rate): Measures the proportion of correctly predicted positive instances among all actual positive instances. It answers: "Of all actual positive items, how many did the classifier correctly identify?"

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

- F1-Score: The harmonic mean of Precision and Recall. It provides a single score that balances both precision and recall, especially useful when there is an uneven class distribution.

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

#### 4.5.3 Fitness Function Calculation

As described in Section 3.2.2, the fitness function combines classification accuracy with a penalty for the number of selected features. The calculation for a given chromosome C (which represents a specific subset of features) involves:

1. Feature Subset Selection: Based on the binary encoding of C, identify the features to be used. For example, if  $C=[1,0,1,1,0]$ , features 1, 3, and 4 are selected.
2. Classifier Training: Train a chosen classifier (e.g., SVM, k-NN) using only the data corresponding to the selected features from the training set.
3. Validation Performance: Evaluate the trained classifier's performance (e.g., Accuracy) on the validation set. This  $\text{Accuracy}(C)$  value is then used in the fitness formula.

4. Feature Size Calculation: Count the number of '1's in the chromosome C. This is  $\text{Size}(C)$ .
5. Fitness Value Computation: Apply the formula:

$$\text{Fitness}(C) = \alpha \times \text{Accuracy}(C) - \beta \times \text{Size}(C)$$

The values for  $\alpha$  and  $\beta$  are hyper-parameters that need to be tuned based on the specific problem and desired trade-off between accuracy and feature count.

#### 4.5.4 Genetic Algorithm Parameter Calculations (Probabilistic)

While not direct output calculations, the following parameters govern the probabilistic operations within the GA:

- Population Size (P): The number of chromosomes in each generation. This is a fixed integer value.
- Number of Generations (G): The total number of iterations the GA will run. This is a fixed integer value.
- Crossover Probability ( $P_c$ ): The probability that two selected parent chromosomes will undergo crossover to produce offspring. Typically,  $P_c \in [0.6, 0.9]$ . If a random number  $r \in [0, 1]$  is generated and  $r < P_c$ , then crossover occurs.
- Mutation Probability ( $P_m$ ): The probability that a bit in a chromosome will be flipped. This is usually a very small value, e.g.,  $P_m \in [0.001, 0.01]$ . If a random number  $r \in [0, 1]$  is generated for each bit and  $r < P_m$ , then that bit is flipped.

These probabilities dictate the exploration and exploitation balance of the GA, influencing its convergence speed and ability to find optimal solutions.

#### 4.5.5 Computational Cost Measurement

The computational expense of a Genetic Algorithm (GA) is primarily assessed through its **time complexity** and **space complexity**. Time complexity refers to the total time required for the GA to find a solution, which is heavily influenced by the number of generations and the time spent in each generation on tasks like extracting feature subsets for every potential solution (chromosome), training and validating a classifier with these features, and executing genetic operations such as selection, crossover, and mutation. Meanwhile, **space complexity** quantifies the memory needed to store the population of chromosomes and the large volume of feature data. Thoroughly considering these computational aspects is crucial for both efficiently implementing a GA and accurately measuring its effectiveness in image analysis feature extraction.

## 5.Future Plans

Based on the successful implementation of this project, the following avenues are recommended for future development and research:

**Core Algorithm and Fitness Function Refinement:** The primary next step is to refine the foundational Genetic Algorithm code for enhanced efficiency and modularity. This includes experimenting with different fitness function formulations to better balance the trade-off between classification accuracy and the cardinality of the selected feature set, ensuring a more robust problem-solving capability.

**Scalability and Dataset Expansion:** The current framework will be tested on larger and more diverse datasets. This will involve analyzing the GA's performance and scalability as data complexity increases and optimizing the solution to maintain high performance on more challenging, real-world data.

**Hybrid Metaheuristic Models:** A key future goal is to develop a hybrid algorithm by integrating the GA with a local search technique, such as Particle Swarm Optimization (PSO) or Simulated Annealing. This approach aims to leverage the GA's global exploration with the fine-tuning and rapid convergence of other methods to find superior feature subsets.

**Multi-Objective Optimization:** We plan to transition from a single, weighted fitness function to a multi-objective approach using algorithms like NSGA-II. This will allow for the simultaneous optimization of conflicting objectives—such as maximizing accuracy, minimizing feature count, and reducing computational time—to generate a Pareto front of optimal trade-off solutions.

**Integration with Deep Learning Architectures:** A more advanced plan involves using the GA to optimize deep learning models. This includes applying the GA to select the most salient features from the intermediate layers of pre-trained Convolutional Neural Networks (CNNs) or to perform Neural Architecture Search (NAS) for designing efficient, problem-specific network structures.

**Real-Time System Optimization:** For practical applications, we will investigate the parallelization of the GA's fitness evaluation process using GPU computing or distributed frameworks. The objective is to significantly reduce the computational time required to identify the optimal feature set, making the system viable for real-time scenarios.

## 6. References

- Al-Ani, A., & Al-Dahoud, A. (2009). Genetic algorithm for feature selection in facial recognition. *Journal of Computer Science*, 5(2), 127-132.
- Bay, H., Tuytelaars, T., & Van Gool, L. (2006). SURF: Speeded Up Robust Features. In the *European Conference on Computer Vision (ECCV)* (pp. 404-417).
- Bhandari, A. K., Kumar, A., & Singh, G. K. (2018). Modified K-means algorithm for image segmentation using genetic algorithm. *Expert Systems with Applications*, 107, 1-13.
- Dalal, N., & Triggs, B. (2005). Histograms of Oriented Gradients for Human Detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)* (Vol. 1, pp. 886-893).
- Lowe, D. G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2), 91-110.
- Tan, P. N., Steinbach, M., & Kumar, V. (2005). *Introduction to Data Mining*. Pearson Addison Wesley.
- Zhang, Y., Wu, L., & Wang, S. (2010). A genetic algorithm for feature selection in MRI brain image classification. *Journal of Medical Systems*, 34(6), 1019-1027.
- Anai, T. A., Al-Hashimi, M., & Anaee, M. A. (2023). Effect of Genetic Algorithm as a Feature Selection for Image Classification. *Iraqi Journal of Science*, 64(11), 6001–6012.
- Feng, G. (2024). Feature selection algorithm based on optimized genetic algorithm and the application in high-dimensional data processing. *PLoS ONE*, 19(5), e0303088.
- Feng, G., et al. (2023). Genetic algorithm-based feature selection with manifold learning for cancer classification. *BMC Bioinformatics*, 24, 159.
- Wang, H., Bai, L., & Yang, N. (2014). Pattern Recognition Application of Improved SURF Algorithm in Mobile Phone. *Applied Mechanics and Materials*, 610, 471–476.
- Bostanci, E. (2015). Enhanced image feature coverage: Key-point selection using genetic algorithms. *arXiv preprint arXiv:1512.03155*.