Mathematical Biosciences and Engineering

MBE, 18(5): 6178–6197.
DOI: 10.3934/mbe.2021309
Received: 07 April 2021
Accepted: 08 June 2021
Published: 14 July 2021

http://www.aimspress.com/journal/MBE

Research article

Fruit-Fly optimization based feature integration in image retrieval

Pavithra Latha Kumaresan[1], Subbulakshmi Pasupathi[1], Sindhia Lingaswamy[1], Sreesharmila Thangaswamy[2], Vimal Shunmuganathan[3] and Danilo Pelusi[4,*]

[1] School of Computer Science and Engineering, Vellore Institute of Technology, Chennai Campus, India
[2] Department of Information Technology, SSN College of Engineering, Chennai, India
[3] Department of Computer science and Engineering, Ramco Institute of Technology, Tamilnadu, India
[4] Faculty of Communication Sciences, University of Teramo, Via Balzarini, 1, 64100, Italy

* **Correspondence:** Email: dpelusi@unite.it; Tel: +390861266036.

Abstract: The content-based image retrieval (CBIR) system searches and retrieves the similar images from the huge database using the significant features extracted from the image. Feature integration techniques used in the CBIR system assign static weights to each feature involved in the retrieval process that gives a smaller number of similar images as a result. Moreover, the retrieval time of the CBIR system increases due to the entire database search. To overcome this disadvantage the proposed work introduced a two-level searching process in the CBIR system. The initial level of the proposed framework uses the image selection rule to select more relevant images for the second-level process. The second level of the framework takes the proposed dominant color and radial difference pattern details from the query and selected images. By using color and texture features of the selected images, similarity measure is calculated. The proposed work assigns optimal dynamic weight to the similarity measure of color and texture features using the fruit fly optimization algorithm. This improves the retrieval performance of the CBIR system.

Keywords: content based image retrieval; dominant color descriptor; radial difference pattern; fruit fly optimization; similarity measure

1. Introduction

Recent advancements in image sensing devices help in acquiring a massive amount of digital images in the fields such as art, medical, satellite, defense, surveillance, web application, etc. In CBIR the specific image search on the huge digital image collections is a difficult task when the user

query is in image form. The CBIR system represents the query and database images using the key features like chrominance intensity, texture, and shape details of the image. The single visual content/feature (i.e., color or texture) based image retrieval gives less number of relevant images as a result [1]. Moreover, the semantic gap between the user query and the relevant results is high. Therefore, two or more visual features are integrated to fill the semantic gap between the original input image and resultant images. The most familiar feature integration in a CBIR system is based on the color and texture features of an image [2–6]. Additionally, texture-shape [7,8], shape-color [9], and color-texture-shape [10–13] details integrated CBIR are developed. However, CBIR systems based on multiple features integration techniques also fail to relate features of the query image with the features of the set of database images. Thus, the CBIR system requires intelligent feature integration techniques to combine extracted features of the image.

1.1. Different types of feature integration techniques in CBIR systems

The different kinds of feature integration techniques in CBIR systems are explained below. The integration of the texture and color details of an image by applying color quantization techniques on the different levels of Discrete Wavelet Transformed (DWT) images [2]. The Discrete Wavelet Transform (DWT) separates low and high-frequency components of the image. The low-frequency component in each level of the decomposed image contains the quality of information into it since the further decomposition levels efficiently utilize it. Each level of DWT images shows the significant details of the image in different resolutions. Here, texture is taken from each level of DWT decomposed image. The Zernike color moments and Contourlet transform texture details are separately extracted from the given image [6]. While integrating these two features, the same weight is assigned for them, which achieved accuracy of around 70% on Wang's database. The integrated color, texture, and shape features of the image by giving equal importance to each feature (i.e., weight of color = weight of texture = weight of shape = 1/3) [8]. The extracted features of an image do not always have equal probabilities for significant information. The above-mentioned characteristic incorporated CBIR structures assigns identical weight to every characteristic of the image, which reduces the accuracy of the retrieval outcome. Therefore, feature selection methods are considered in the retrieval system. Here, the prominent feature vector has high weightage compared to the other features of the image. The feature integration technique [10] gave different weights for the shape and texture features used in the retrieval process. The Angular Radial Transform (ART) based shape feature has higher weight compared to the color difference histogram texture feature obtained from Lab color space. The three different histograms are taken from HSV color space quantized image, different levels of DWT images, and Robinson compass edge filtered images [12]. The feature integration step assigned a weight of 0.5 to the color histogram. Subsequently, texture and shape features histograms were assigned a weight of 0.2 and 0.3, respectively. However the semantic gap stays excessive, while the retrieval accuracy depends on the significant characteristic of the feature and their amount of contributions in CBIR. The amount of each feature contribution in feature integration is not static, and it changes dynamically according to the image. The optimal weight assignment methods improved the retrieval accuracy of the CBIR system. Therefore, an optimal weight assigned for extracted features such as dominant color, Frobenius form of DWT images, and curvelet transform shape details [13]. The weight of each feature is dynamic, based on prominence of a feature in an image. However, the performance of feature integration technique

based on the optimal weight is affected by the type of features involved in integration. Subsequently, retrieval time taken by these techniques will depend on the total number of images involved in retrieval and length of the feature vector. Hence, CBIR system needs the optimal image search space as well as the optimal weight for each instance involved in the retrieval process. The database images and query images use the same feature extraction method for color and texture. In convolutional neural network-based feature extraction, each database needs different layers of the convolutional network to get an optimal feature map. Those optimal feature maps are obtained from the final activation layer which gives the highest accuracy in image classification/recognition. Instead of building the CNN from the scratch, transfer learning is also widely used in classification. However, fine-tuning of parameters in the CNN model is time-consuming and database-dependent. Moreover, feature maps of CNN are effectively handled by the classification model rather than the similarity measure and optimization algorithm [14]. Moreover, CNN needs a considerable amount of images to perform training and testing over them. Otherwise, it results in over-fitting which is the biggest problem in image classification [15,16]. Hence, this work didn't consider the CNN feature map for the proposed image retrieval work. The hybrid CBIR [17] system uses CNN classification model in the first phase and then it uses the computer vision technique called SURF to match the similar images with the help of classification results. Here the databases are created by taking different view (Image) of the same object. If classification fails to give the correct category of image then the object are cannot identified and retrieved correctly from the database. The SURF features descriptor approximately match different images (taken from different orientation and position) of the same object. However, SURF feature vector for the images of different objects from the same category has less accuracy in similar image retrieval.

Many kinds of optimization techniques are mimicked from the food search behavior of insects/birds/animals (i.e., Harris's hawk optimization [18], Moth search algorithm [19], Fruit Fly Optimization [20]) to provide solution for the real word problems. Some of the real world problems are parameter estimation for frequency-modulated (FM) sound waves, Messenger: Spacecraft Trajectory Optimization Problem, Large Scale Transmission Pricing Problem, etc. Each optimization algorithm must have the fitness function in it. The problem type considered in the optimization has to decide whether it needs to maximize or minimize the fitness function. According to the requirements, parameters of the optimization algorithm get tuned and give optimal solution. The Monarch butterflies migration behavior during the summer/autumn time is imitated to minimize the fitness function value of the continuous optimization problem [21]. Its gives optimized solution for the high dimensional parameters tuning problems. Slime mould algorithm is introduced to produce solution for stochastic optimization problem in food search. This algorithm produce adaptive weight to produce positive and negative feedback for food search and it is experimented over the welded beam structure problem, pressure vessel structure problem, etc. [22]. The hunger game search algorithm is introduced to give optimized performance in the field of artificial intelligence and machine learning [23]. Instead of introducing new optimization algorithms many authors has investigated the existing optimization algorithms and identified the problem associated with them. This been an active research area and suggested the solution to overcome the problems associated with them. Therefore, many variants of original optimization algorithms are also available. The fruit fly optimization algorithm imitates the food searching behavior of them. Here the location of each fruit fly is updated randomly and converge fast towards the local minima when the optimization is required for high dimensional data. To reduce the issues raised in location update and

handling high dimensional data handling, bat sonar strategy and combination of Gaussian distribution with student distribution are introduced [24]. Meanwhile, the global search capability of the FFO is improved by the Bare fruit fly optimization algorithm (BareFOA). The BareFOA includes dynamic step length for tuning the population and Gaussian bare-bones mechanism to avoid the local minima [25]. Each fruit fly is combined with artificial fish swarm algorithm, to reduce the multi-population problem [26]. Likewise, sine cosine algorithm is used to update the location details of each fruit fly to get the global optima for the dimensional data [27]. Recently, the random search strategy of the FFA is replaced with the whale-optimization algorithm this will able to handle large amount of search space effectively [28].

Firstly, the proposed work selects the optimal subset of images. This subset selection process is responsible for creating an optimal search space regarding the query image. The lower and upper limit of the subset selection process is estimated from the average and standard deviation value of each color channel of the query image. Next level of the proposed CBIR, consider the key details, mainly dominant color and radial difference texture patterns from the selected and query images. Finally, the proposed work uses the Fruit Fly Optimization (FFO) technique to maximize the precision of the retrieval result by assigning the optimal weights for each feature considered here. The weights are dynamically changing based on the query image given by the user.

The overall structure of this paper is given as follows: the proposed optimized feature integration is explained in Section 2; Section 3 shows the experimental results and discussions of the proposed CBIR system. Section 4 gives the conclusion of the work carried out in the proposed work.

2. The proposed hybrid feature based CBIR framework

Figure 1 shows the work flow of the proposed CBIR framework. Initially (first) the proposed work performs the image selection over the huge image database. The selection rule takes the average and deviation details of the query image and database images. The next level (i.e., second level) of the proposed work gives the optimal combination of significant features taken from the color and texture feature. Figure 1 differentiates the initial level from the second level process of the proposed work by orange and green colors. The number above each arrow line indicates the workflow of the proposed CBIR framework.

2.1. First level of the proposed framework

The color mean and deviation details are taken from the image in the form of query and database. The following section explains the processes carried out in the proposed CBIR framework.

2.1.1. Color mean and standard deviation

Average value of Red (R), Green (G), and Blue (B) color channels of each image is extracted from the Eq 1. The proposed work groups the average value of above said color channels of the images from the database by applying K –means clustering algorithm. The number of clusters formed over the average values of the database images must be high enough to hold the variations present in them. Alternatively, the proposed work uses the Eqs 1 and 2 to estimate the average and standard deviation information of the pixels present in the query image. The Standard Deviation (SD)

gets minimized if the pixel values present in an image are close to its own mean value. If SD gets a hike then it implies, maximum number of pixel values are not near with the mean value.

$$Average(CC_{ch}) = \frac{1}{r \times c} \sum_{i=1}^{r} \sum_{j=1}^{c} CC_{ch}(P_{ij}), ch = \{\mathrm{Red} \mid Green \mid Blue\} \tag{1}$$

$$SD(CC_{ch}) = \sqrt{\frac{1}{r \times c} \sum_{i=1}^{r} \sum_{j=1}^{c} \left(CC_{ch}(P_{ij}) - Average(CC_{ch})\right)^2}, ch = \{\mathrm{Red} \mid Green \mid Blue\} \tag{2}$$

where, *ch* denotes the color channel. *r* and *c* represents the row and column of each color channel image CC_{ch}. $CC_{ch}(P_{ij})$ denotes the intensity in the i^{th} row and j^{th} column of the image.

2.1.2. Formation of image selection rule

Image selection step uses the average and standard deviation of query image to form the selection rule. The selection rule has low and high limit in it. These two limits are derived from Eqs 3 and 4 [29]. The set of database images falls between the low and high limits are selected for the next level of the process.

$$LL(CC_{ch}) = Average(CC_{ch}) - Std(CC_{ch}) \tag{3}$$

$$HL(CC_{ch}) = Average(CC_{ch}) + Std(CC_{ch}) \tag{4}$$

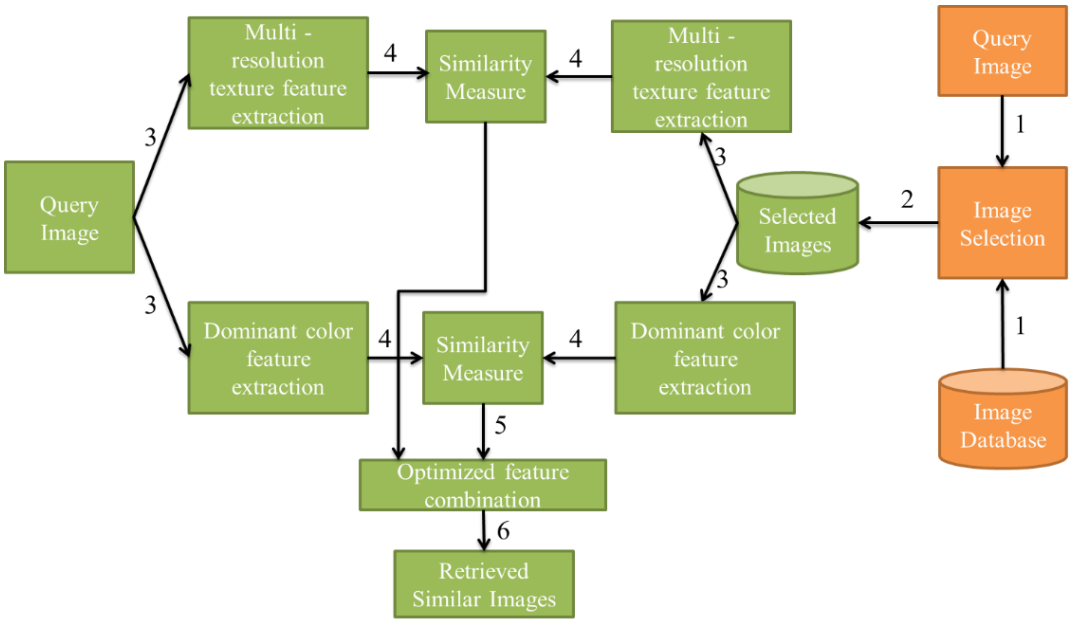

Figure 1. The proposed features integrated CBIR framework.

2.2. Second level of the proposed framework

The second level of the proposed framework considers features of the query and set of images selected from the initial level of this framework. The significant features of images considered in this process are estimated from the dominant color and radial difference of local binary patterns. The below sections give the details of the parameters involved in the proposed color and texture feature extraction process.

2.2.1. Dominant color feature

HSV's individual color channel will be segregated into various groups; because the information about color will be distributed non uniformly in HSV. Channel H clutches additional details about the colors and it has a comparatively higher number of clusters when compared to S and V channels. Mostly, this dominant color feature depends on the number of clusters that's used for representing the color. With the available clusters on 'H', the optimal number has been chosen to know about the available dominant color details. At the outset, to represent 'a dominant color information this work recommends to form 8 clusters. Equation 5 gets the channel's seed information. The number of clusters formed on the H color channel is by taking the maximum value over the difference of eight the Calinski-Harabasz index obtained from them. The mean values of the resultant optimal clusters are combined to form seed points. Using these seed points, dominant color clusters will be formulated by K-means clustering approach.

$$\text{Seed\_Points}(c) = \frac{I_{MEAN}}{TNC} + (c-1) \times \frac{I_{MAX}}{TNC} \tag{5}$$

$$Calinski\text{-}Harabasz_{index} = \frac{\text{Between class variance}}{\text{K-1}} \times \frac{M-K}{\text{Within class variance}} \tag{6}$$

$$\text{Within class variance} = \frac{1}{N_i} \sum_{n \in C_i} \sum_{i=1}^{K} \left\| n - C_i \right\|^2 \tag{7}$$

$$\text{Between class variance} = \frac{1}{K} \sum_{i=1}^{K} \left\| C_i - \bar{C} \right\|^2 \tag{8}$$

where, M represents the total number of pixels available in the image. N_i represents the total number of pixels exists in the i^{th} cluster. C_i is the i^{th} cluster center value. $\bar{C}$ is global center of the image. n indicates the pixel in the dominant color cluster. The mean and maximum value of pixel value of the image is denoted by I_{mean} and I_{max}. TNC represents the total number of clusters. C denotes the number of clusters varying from 1 to TNC.

Then, the extracted dominant colors of the query image and the set of images selected from the initial level of the proposed framework are taken for similarity measure calculation. The similarity measure suggested in [30] has high retrieval results. So, it is used to calculate the distance between the dominant colors in the query image and set of selected images.

*2.3. Radial difference local binary patter*n

The radial difference LBP takes the positive and negative sign details between the equally spaced circular points on two different radii with the same radial direction. It takes the patterns of sub image from changing radius and number of pixels in the circular neighborhood. The RD_LBP feature extraction is illustrated in Eq 9.

$$RD\_LBP_{p,r,r-\delta} = \sum_{k=0}^{p-1} s\left(I_{k,r} - I_{k,r,r-\delta}\right)2^k, s(d) = \begin{Bmatrix} 1, d \geq 0 \\ 0, d < 0 \end{Bmatrix} \tag{9}$$

where *p* indicates the number of equal distanced pixels on the circular neighborhood, r is the radius between the center pixel and the points on the circular neighborhood. The $r-\delta$ gives another radius value from which the circular neighborhood pixel difference is calculated. The amount of displacement is derived from the value of δ. The RD_LBP does not take the center pixel for estimating the patterns in the sub image this affects discriminant pattern formation. The RD_LBP value is calculated from the eight equally spaced pixel values with same radial direction on radii 2 and 1. The D1- distance measure is used to give more number of relevant images for the texture feature-based CBIR.

2.4. Normalization

The similarity measure estimated between the features of the query image and small set of images selected from the initial level of proposed framework. The values present in the feature vectors (Color and texture) are not in the same range. For example, texture feature vector values vary from 0 to maximum number of times the pixel occurs in the image, while color features vary between 0 and 1. Thus, the similarity measure values of proposed features are varying in an unbounded fashion. So we need to normalize to restrict the maximum and minimum variations available in the similarity measure values to a specific range. The proposed derived framework uses the Min-Max normalization [31] on the similarity measure values of the recommended dominant color and radial difference local binary pattern. The Min-Max normalization is implemented through Eq 9 which limits the values obtained in the similarity measure between the ranges 0 to 1.

$$ND(i) = \frac{D1_f(i) - \min(D1_f)}{\max(D1_f) - \min(D1_f)}, i = 1,2,..., L \tag{10}$$

where, *L* is overall count of selected images. $D1_f(i)$ represents the similarity measure value of the i^{th} image in the set of images selected for the second level of the proposed frame work. $\min(D1_f)$ and $\max(D1_f)$ indicate the minimum and maximum value of the similarity measures obtained from the proposed color and texture features of the query image and set of selected database images.

Then, the normalized color and texture distance measures values are fused to give a new similarity measure for the set of images participating in the second level of the proposed framework. The new similarity measure is calculated from Eq 10.

$$New\_SM(i) = w1 * ND\_CF(i) + w2 * ND\_TF(i),\ i = 1,2,..., L \tag{11}$$

where, L is the overall count of selected images. New_DF total number of selected images. New_SM represents the new similarity measure of the i^{th} image in the set of images involved in the second level of the proposed work. $ND\_CF(i)$ and $ND\_TF(i)$ indicate the normalized distance measure of color and texture features of the i^{th} mage in the set of involved in the second level of the proposed work. $w1$ and $w2$ are the optimal weights of each similarity measures (i.e., $w1 + w2 = 1$).

2.5. Fruit fly optimization in feature integration

The optimal weight assignment process of the proposed framework is carried out with the help of fruit fly Optimization (FFO) algorithm. The FFO algorithm emulates the properties of fruit flies during food search [32,34–37]. The variants of FFO are introduced to overcome the problem of handling multidimensional data and large search space of original FFO. In the proposed work, the search space size is reduced and dynamic because optimization is applied between the query and resultant images of the first level of this proposed work. The dimensionality of the proposed work is two. Therefore, the efficient unmodified FFO is considered in this work. The proposed framework follows the FFO algorithm to select the optimal weight for the normalized similarity measures of two features (dimension = 2). These two optimal weighted similarity measure values are added to give the new similarity measure. The search space of optimal weight is in the range of 0 to 1. The weight of each feature's normalized similarity measure value is modified according to the value of fitness function selected in the proposed framework. The retrieval performance is modified based on the optimal weight of each feature similarity measure. The performance metric named "average retrieval precision" quantitatively measures the correctness of the retrieved result. Thus, the proposed framework selects the average retrieval precision as the fitness function for the FFO algorithm, and it converges at a high precision rate.

Figure 2 illustrates the optimal weight selection process carried out in the proposed work. The optimal weight for the proposed color and texture information is estimated using the set of fruit flies involved in the FFO algorithm. The dimension of each fruit fly in the FFO algorithm is assigned as two (i.e., Dominant color and Radial difference texture features). The algorithm has the number of randomly positioned fruit flies on the search space. The search space has a set of possible solutions for the optimization problem. The range of search space is varying from 0 to 1. The optimal weights obtained from the search space are responsible for improvement in the average retrieval precision.
Each fruit fly of the FFO algorithm includes its initial location information ($X_{initial}$, $Y_{initial}$) and number of iterations. Then each of them tries to find the food location by travelling random distance from the initial location (Eq 12).

$$X_{search}(\text{i})= X_{initial} + Random\_dist; \quad Y_{search}(\text{i})= Y_{initial} + Random\_dist; \tag{12}$$

where, $Random\_dist$ takes the value from 0 to 1.

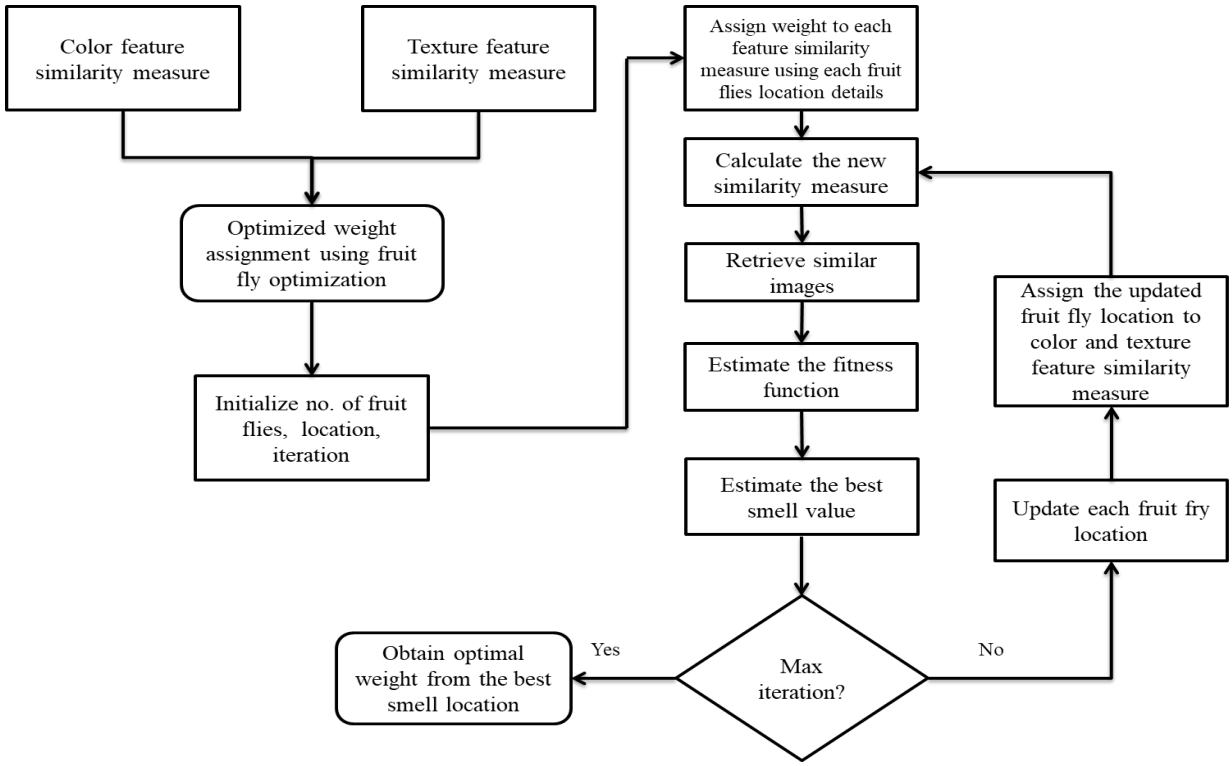

Figure 2. Optimal weight selection using FFO algorithm.

The food smell concentration for each fruit fly location is judged from the distance information calculated from the origin. Here, judged smell concentration and distance information are inversely proportional to each other.

$$SJ(i) = \frac{1}{Dist_i} \tag{13}$$

$$Dist_i = \sqrt{(X_{search}(i))^2 + \ (Y_{search}(i))^2} \tag{14}$$

Apply the judged smell concentration value ($SJ(i)$) of every fruit fly and evaluate the fitness function.

$$smell_i = fitness\_function(SJ(i)) \tag{15}$$

Find the fruit fly that has maximum smell concentration (i.e., $\max\{smell_i\}$) and extract its corresponding x and y coordinate value. Then, update the $X_{initial}$ and $Y_{initial}$ values with new best x and y coordinates detail as shown in Eq 16.

$$X_{initial} = best\_X;\ Y_{initial} = best\_Y \tag{16}$$

The position of $X_{initial}\ and\ Y_{initial}$ is estimated after each iteration. The value of $X_{initial}\ and\ Y_{initial}$ is updated at the end of each iteration if the value of current smell is greater than the previous smell otherwise no update is needed. This process is repeated until it meets the maximum number of iteration. Once it reaches the maximum number of iteration the optimization algorithm returns the best smell detail. It gives a high retrieval precision rate of the proposed CBIR framework. The location detail associated with best smell gives the optimal weight assigned for the similarity measure of the proposed color and texture features.

3. Experimental results and discussions

The feature extraction methods and retrieval experiments involved in this work are implemented in MATLAB R2015a environment along with the dual-core processor, 8GB memory, and 64-bit Windows operating system. The following experiments evaluate the proposed CBIR framework. The proposed framework is implemented on five databases Wang's, Corel -10k, OT-scene, free photo and GHIM. The experiments test the proposed framework by taking 20% of images as query images from each database at 10 different intervals. Later, the proposed system performance is evaluated by taking the average of retrieval precision and recall. The initial level of the proposed framework narrows down the image search space with the help of the image selection rule. This proposed selection rule of the query image, selects the small set database images for the next level of process. This process again checks the mean value of each image in the database, which increases the time of the image selection process. Thus, the K-means clustering algorithm is proposed as a pre-processing step to cluster mean value of database images.

3.1. Pre-processing

The mean value of 80% of the images (i.e., RGB color space images) in each database image is grouped into a different number of clusters. The remaining 20% of images are used to test the proposed CBIR system framework. The initial seed points of the clustering process are estimated from the mean details extracted from the RGB color space images from the overall database using the total number of clusters, mean and maximum values of the mean of database images. The number of clusters formed on the mean values of the database images should maintain the differences available in them.

The pre-processing step of the proposed framework groups the mean values of database images into different numbers such as 2, 3, 4, and half of the total number overall count of images in the database. For example, Wang's database has a total of 1000 images, and 20% of images (i.e., 200 images) from this database act as query images. The remaining 80% of images (i.e., 800 images) are participated to create the search space in a similar image retrieval process. The proposed framework takes the mean value of these 800 images and groups them starting from 2 to 400 different clusters. Table 1 shows the impact of the small numbers of clusters on the mean values of the database images. The number of clusters such as 2, 3, and 4 fail to hold the variation among the mean values of the database images. This is due to the fact that K-means clustering approach with a small number of clusters causes high variance within the cluster. Therefore, the mean values of the small number clusters are not applicable for selecting the set of suitable images to the second level of process (i.e., database images cluster mean values fail to pass the low and upper thresholds used in the image

selection rule). Moreover, the number of images in each cluster formed on the database images deviate from the goal of the initial level process. For K = 2, 800 images of the Wang's database are grouped into two clusters, the first cluster has 691 images, and the second cluster has 109 images into it. The mean value taken from the 691 and 109 images are checked with the threshold value obtained from the query image in the image selection rule. The cluster mean of the 691 images lies between the mean and standard deviation of the query image. Then the cluster which has 691 images is selected by the image selection rule of the proposed framework. This has a higher search space compared to the clusters formed based on the high value of K, for the same query image. If both clusters mean values do not pass the image selection rule, no images are selected for the next level of process, which is not true for high value of K since a large number of clusters maintain the variation present in data points considered for the clustering process. Thus, the proposed work uses a large number of image clusters in the pre-processing stage which selects more suitable images for the second level of the proposed image retrieval process.

Table 1. Impact of a small number of clusters on the mean values of database images.

Sl. No.	Database Images	K = 2	K = 3	K = 4
1	**Free Photo**	1500,1270	796,704,1270	865,557,362,986
2	**GHIM**	3563,4437	3609,2435,1956	2967,2337,1154,1542
3	**Wang's**	691,109	382,306,112	187,183,105,325
4	**Corel-10k**	2888,5112	1395,2903,3702	1727,2456,1330,2487
5	**OT-Scene**	1170,980	547,601,1002	616,404,542,588

3.2. Impact of the image selection rule

Image selection rule is used to minimize the count of images involved in the proposed framework of the CBIR system. The image selection rule checks the mean values of 400, 1385, 4000, 4000, and 1075 clusters formed on color channels of 80% of images in free photo, GHIM, Wang's, Corel-10k, and OT-scene databases respectively. The image selection rule checks whether the mean value of each cluster lies between the upper and lower threshold used in this rule. The mean of the cluster which satisfies this rule passes the set of images (i.e., new subset database) associated with that cluster to the second-level process. Here, the search space is narrowed down in maximizing CBIR system retrieval speed. The benefits of large numbers of clusters on the mean value of database images are studied by applying image selection rules over the original mean values (i.e., without preprocessing), a small, and a large number of clusters formed on that mean values. The experiment takes 10 query images from the OT-scene databases. The threshold for image selection rule is estimated from these query images. Then, the original mean, small, and large numbers of clustered mean values of the images are passed over the image selection rule. The number of images selected for the next level of process is given in Figure 3. For the second level of process the original mean of each image and cluster number equal to half of the total number of images participated in the retrieval process has given an approximately equal number of images. The preprocessing steps increase the clustering process time when K is high. Whereas the small number of cluster mean (i.e., K = 2) based image selection process does not select enough number of images for the second level of process. The mean value from the cluster number two fails to select relevant images from the database images for the further level of process. (i.e., when K = 2, the mean value of these clusters

does not select any images from the database for images 2, 5 and 8). Moreover, a large number of clusters reduce the number of comparisons between the query and database images by half compared to the original mean value based image selection. Thus, this work selects K as half of the overall count of images in the databases.

Since the threshold of selected image is fully dependent on the mean and standard deviation of the image, the new subset database image will randomly differ. Further, the size of the selected image subset is not predictable. In the final stage, distance of related images will be minimized; fairly the distance will be lesser than the initial stage's distance. The initial level of proposed work takes the total number of images for image selection (i.e., 3162, 10000, 1000, 10000, and 2688), and it approximately selects 1762, 3462, 497, 3500, and 1662 images from the free photo, GHIM, Wang's, Corel-10k, and OT-scene databases, respectively. Then, it calculates the similarity measure between the query image and the approximately selected images. This increases the retrieval speed of the CBIR system. Figure 4 represents the amount of images retrieved from the CBIR system with and without the image selection rule. Figure 4 illustrates the CBIR system, which does not have the image selection rule. Similarity is calculated by taking all images of the query and database. Therefore, the retrieval time of that CBIR system is high, compared to the proposed image retrieval system. The average retrieval time of the proposed (without FFO) and traditional CBIR system for 20% of query images are tabulated in Table 2. Moreover, the mean and standard deviation combination takes less amount of time to select the pertinent images from the database images. Hence, it is placed on the initial level of the proposed framework. The average retrieval time of the image selection process is tabulated in Table 2.

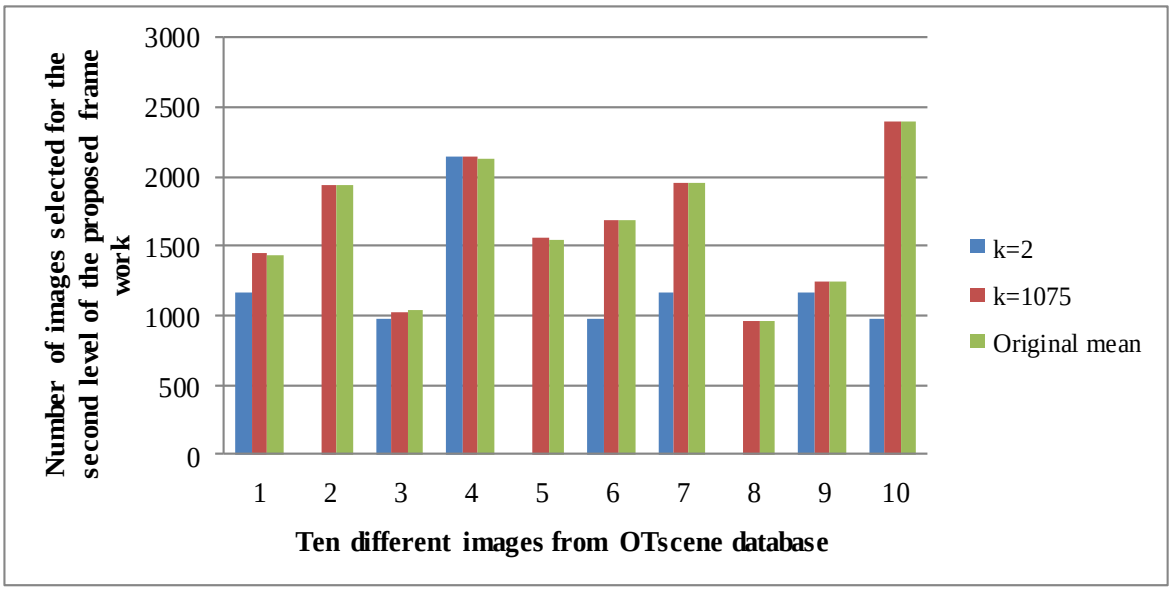

Figure 3. Impact of large number of clusters used in image selection.

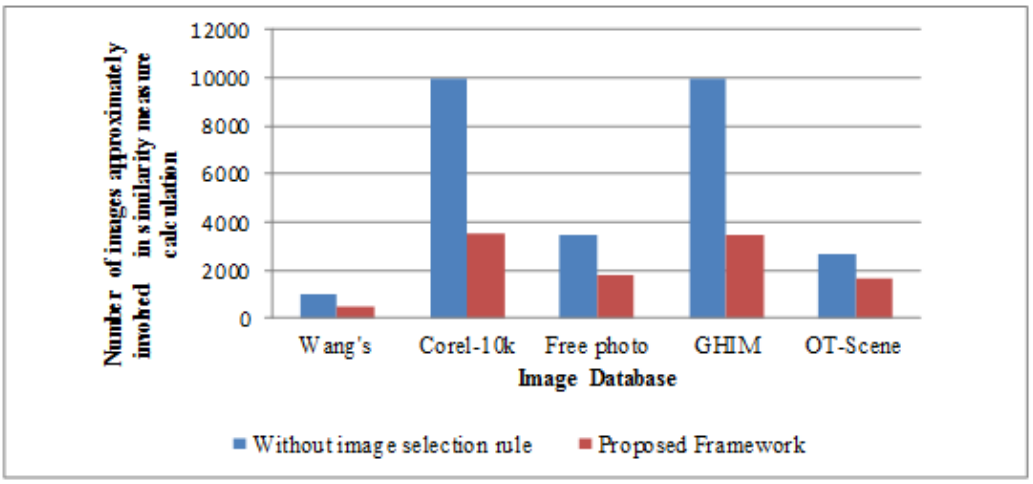

Figure 4. Efficiency of image selection rule in the proposed CBIR framework.

Table 2. Image retrieval time analysis.

Image retrieval time (s)		Free photo	GHIM	Wang's	Corel-10k	OT-Scene
Without image selection rule (TM)		22.34	44.68	6.25	45.68	18.42
With image selection rule (PM)	1st level of the process	0.032	0.061	0.003	0.068	0.070
	2nd level of the process	2.483	8.246	1.069	7.246	2.56
	Total	**2.515**	**8.307**	**1.072**	**7.314**	**2.63**

TM- Traditional Method, PM- Proposed Method

3.3. Performance analysis of feature integration techniques

The proposed color and texture features are integrated in three different ways, and the performance associated with them is evaluated by precision and recall metrics of the CBIR system. By using the dominant color extraction method, the first step of the feature integration approach clusters the dominant colors available in the image. Then, the proposed texture feature extraction method is applied on the dominant color clustered image (i.e., radial difference binary patterns are extracted from the dominant color formed on the HSV color space image). Then, the D1-similarity measure is calculated between the dominant color texture feature of the query and selected images. The feature integration technique using the dominant color based texture feature extraction method has the lowest performance in image retrieval. The dominant color-based image quantization method does not highlight the small variations present in the image. Thus, the radial difference local difference binary patterns of the dominant color image lack in the ability to extract the local pixel difference details of the image. These patterns are inadequate to differentiate the two different images, which affect the retrieval performance of the CBIR system. In the second method, the integration technique gives equal weight for the proposed color and texture features similarity measures. The color and texture features individually use different similarity measures. Thus, these two features similarity measures give different distance values for each image. An assignment of equal weight for each feature vector will affect the performance of the CBIR system. The CBIR system without optimal weight assignment has less retrieval rate. Table 3 shows the top 20 average retrieval

precision and recall values for the query images used for testing the proposed CBIR system. The texture feature based image retrieval system averagely gives 52.92%, 62.05% and 66.9% similar images over the corel-10k, GHIM and OT-Scene database images. These retrieval results are 0.71%, 0.4% and 2.6% higher than the multiple features integrated CBIR system which gives equal weight for the proposed color and texture features extracted from the Corel-10k, GHIM and OT-Scene databases. The performance analysis of the proposed color and texture features integrated CBIR system using equal weight for each proposed feature gives 79.5% and 75.36% average precision over the Wang's and the free photo databases. Whereas the third method assigns an optimal weight for the proposed features similarity measures using the FFO algorithm. The FFO algorithm used in this experiment takes the number of fruit flies to be 20, and the number of iterations to be 30. After each iteration, the best smell concentration and its corresponding locations are updated. The best smell value is considered as the optimum precision rate, and its parallel two-dimensional fruit fly position information acts as a weight for the color and texture features similarity measures, respectively. The variant of FFO algorithm (SCA-FFO) is implemented for experimental analysis and got 64.97% retrieval precision which is 1.97% is higher than the original FFO. The retrieval result of Monarch butterflies Optimization (MBO) algorithm gives retrieval results better than PSO based image retrieval. However, the retrieval results of MBO is 5.08% (Wang's), 3.75% (Corel-10k), 4.14% (Free Photo), 2.22% (GHIM) and 5.46% (OT-scene) less compared to the retrieval results obtained using FFO algorithm over those databases. The particle Swarm Optimization (PSO) algorithm with two dimensions, 20 particles, and total number of iteration as 30 also tested over the five databases. Due to its slow motion towards the optimal solution this optimization technique requires more iteration. Thus it gives less number of similar images compared to the FFO based CBIR system. The retrieval precision of the FFO algorithm is depicted in Figure 5. This clearly tells that integration using optimal weight has a maximum impression than the CBIR of individual based. Thus, FFO-based feature integration system has achieved a high retrieval rate of 83.67%, 74.38%, 86.98%, 63.18%and 74.32% over free photo, GHIM, Wang's, Corel-10k, and OT-scene databases, respectively, in the CBIR system.

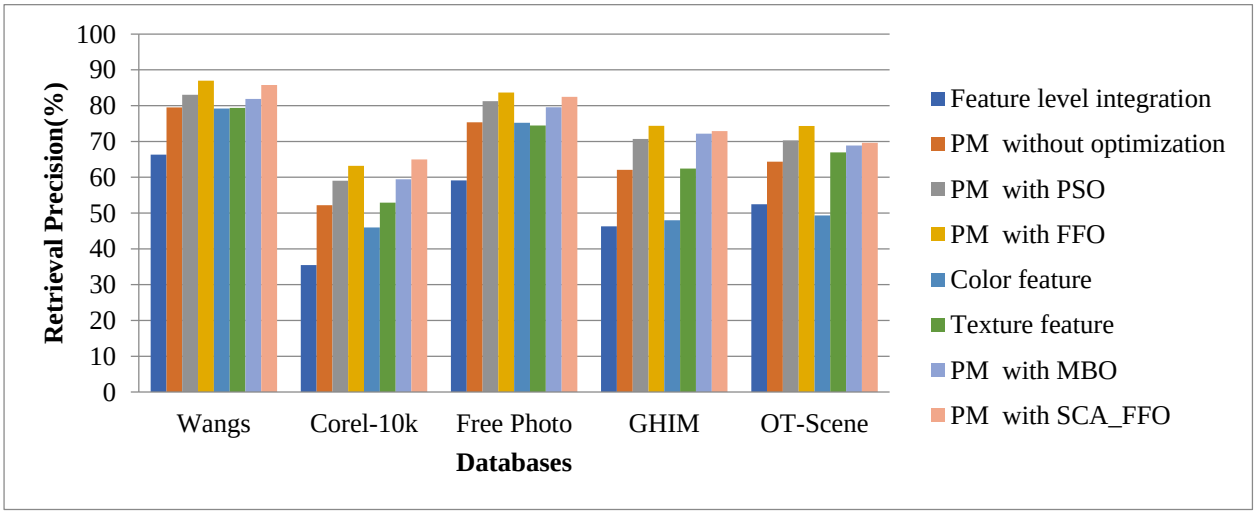

Figure 5. Performance analysis of the CBIR system using the proposed individual feature and multiple features.

Table 3. Performance analysis of proposed and individual.

Database		Feature level integration	PM without optimization	PM with PSO	PM with FFO	PM with MBO	PM with SCA_FFO	Color feature	Texture feature
Wang's	AP (%)	66.32	79.5	83.06	**86.98**	81.9	85.76	79.2	79.36
	AR (%)	13.26	15.9	16.61	**17.40**	16.38	17.15	15.84	15.87
Corel-10k	AP (%)	35.47	52.21	59.03	**63.18**	59.43	64.97	45.97	52.92
	AR (%)	7.09	10.44	11.81	**12.64**	11.89	12.99	9.194	10.58
Free photo	AP (%)	59.12	75.36	81.25	**83.67**	79.54	82.45	75.23	74.44
	AR (%)	11.82	15.07	16.25	**16.73**	15.91	16.49	15.05	14.89
GHIM	AP (%)	46.28	62.05	70.68	**74.38**	72.16	72.89	48	62.45
	AR (%)	9.26	12.41	14.14	**14.88**	14.43	14.57	9.6	12.49
OT-Scene	AP (%)	52.46	64.35	70.31	**74.32**	68.86	69.65	49.34	66.95
	AR (%)	10.49	12.87	14.06	**14.86**	13.77	13.93	9.87	13.39

Bold value indicates best results. PM- Proposed Method

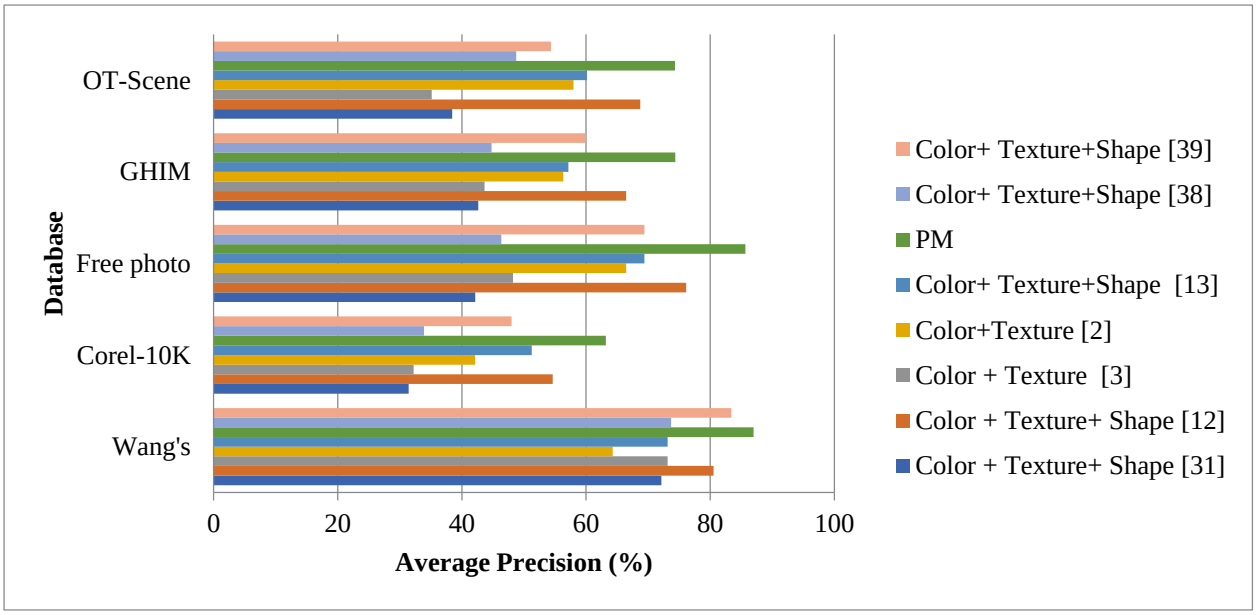

Figure 6. Performance comparison of the existing and proposed features integrated CBIR systems.

3.4. Performance comparison of the multiple feature integrated CBIR systems

The performance of the proposed color and texture features integrated CBIR system is compared with the existing multiple features integrated CBIR systems. The average precision and recall details are evaluated from these multiple feature integrated CBIR systems, and the evaluation results are depicted in Figure 6. The average precision and recall values associated with the retrieval results shown in Figure 6 is tabulated in Table 4. The retrieval experiments estimate the performance of each retrieval system using 10 different sets of images (20% each) taken from the database. The retrieval time associated with each CBIR system is estimated from the five databases such as Wang's, Corel

-10k, OT-scene, free photo and GHIM. The feature integration [3] method gives equal weight for Euclidean distance measures of the color and texture features. Figure 6 illustrates that the CBIR system which gives equal weight for the color and texture features has lesser precision and recall compared to the CBIR system which integrates multiple features using optimal weights. The optimal weight assignment can be static and dynamic. The static optimal weight assignment technique gives different weights for each feature present in the CBIR system, which is not changed for new query images of the retrieval process. A static optimal weight assigned for the shape and texture details of the image [31]. The optimal weight gives more importance to the shape feature than the texture feature of the image, which is constant for the different kinds of image. Despite the static optimal weight, color details used in selecting the top 30 closely matched images are responsible for the reduced precision and recall rate since the texture and shape features extraction methods are only applied over the top 30 images selected using the color feature. The quantized color histogram of the Haar wavelet transformed approximation and horizontal channels are inadequate to deduce the significant details from the decomposed images. Thus, it gives 66.47%, 56.33%, 64.3%, 42.11%, and 57.98% retrieval precision on free photo, OT-Scene, Wang's, Corel-10k, and GHIM databases, respectively [2]. In [13], the texture and shape features of the image by giving equal weight for them. The texture and shape features are in the form of histogram and it gives third-best retrieval results in comparison with the performance of other CBIR systems. However, the feature integration method [12] assigns an optimal dynamic weight for color, texture, and shape feature in a dynamic manner. Here, the weights of the three features are optimally changed based on the PSO algorithm and a precision of 76.14%, 66.47%, 80.53%, 54.62%, and 68.74% is achieved over the Wang's, Corel-10k, free photo, GHIM, and OT-scene databases. Even though PSO-based integration is preferred, the choice of features for integration is also responsible for getting high performance in retrieval results. Thus, the proposed framework optimally integrates efficient color and texture features of the image to achieve high precision and recall rate for different sets of query images. The optimal combination of the proposed color and texture features gives an average precision of 85.67%, 74.38%, 86.98%, 63.18%, and 74.32%, and for free photo, GHIM, Wang's, Corel-10k, and OT-scene databases, respectively. The significant color- and texture features dynamic optimal combination proposed in this paper has 9.53%, 7.91%, 6.45%, 8.56% and 5.58% higher precision rate on the free photo, GHIM, Wang's, Corel-10k, and OT-scene databases compared to the optimal color, texture, and shape features combination [10]. The bilayer based image retrieval [38] method extracts the Color, Texture, and Shape feature. The first level of the process uses texture and shape features of the query and all database images. The second level estimated the distance between the query and database selected images from the first level. However, un-weighted feature of this method gives less significant result in the retrieval. Due to the misclassified results of the Support Vector Machine (SVM), CBIR system retrieves unrelated images on the top of the retrieval results. Thus it reduces the retrieval precision which is clearly projected in the Figure 6. The retrieval time of the PSO-based feature integrated CBIR system [12] is high compared to the other feature integrated techniques considered in the experiment which is shown in Figure 7. The number of iterations and the number of particles involved in the PSO algorithm are responsible for the increase in retrieval time. The three dimensional 20 particles with 50 iterations are considered for getting the number of relevant results. Even though the proposed method uses PSO algorithm for optimal weight assignment, it takes less amount of time than the CBIR system [12]. This can be achieved by the preprocessing step and the image selection rule of the proposed framework. The preprocessing step reduces the total number of

comparisons by half of its total number of images in the database. Rather than applying optimization over all the database images, the proposed work applied FFO on the color and texture features of the selected images. The retrieval system suggested in [31] takes less time to retrieve similar images from the database since the feature vector has three dimensions in the initial level of the CBIR system to select the top 30 closely matched images. Then, the texture and shape details are extracted from these 30 images and similar images searched within them.

Table 4. Comparison of existing and proposed integrated features – Recall and Average precision.

Database		Color + Texture+ Shape [16]	Color + Texture+ Shape [11]	Color + Texture [3]	Color +Texture [2]	Color+ Texture +Shape [13]	Color+ Texture Shape [38]	Color+ Texture+ Shape [39]	PM
Wang's	AP (%)	72.12	80.53	73.14	64.3	73.15	73.7	83.39	**86.98**
	AR (%)	14.42	16.11	14.63	12.86	14.63	14.74	16.678	**17.40**
Corel-10k	AP (%)	31.41	54.62	32.21	42.11	51.24	33.89	48.01	**63.18**
	AR (%)	6.28	10.92	6.44	8.42	10.25	6.778	9.602	**12.64**
Free photo	AP (%)	42.15	76.14	48.24	66.47	69.4	46.34	69.41	**83.67**
	AR (%)	8.43	15.23	9.65	13.29	13.88	9.268	13.882	**16.73**
GHIM	AP (%)	42.63	66.47	43.64	56.33	57.14	44.78	59.9	**74.38**
	AR (%)	8.53	13.29	8.73	11.27	11.43	8.956	11.98	**14.88**
OT-Scene	AP (%)	38.43	68.74	35.14	57.98	60.14	48.75	54.34	**74.32**
	AR (%)	7.69	13.75	7.028	11.60	12.03	9.75	10.868	**14.86**

PM- Proposed Method

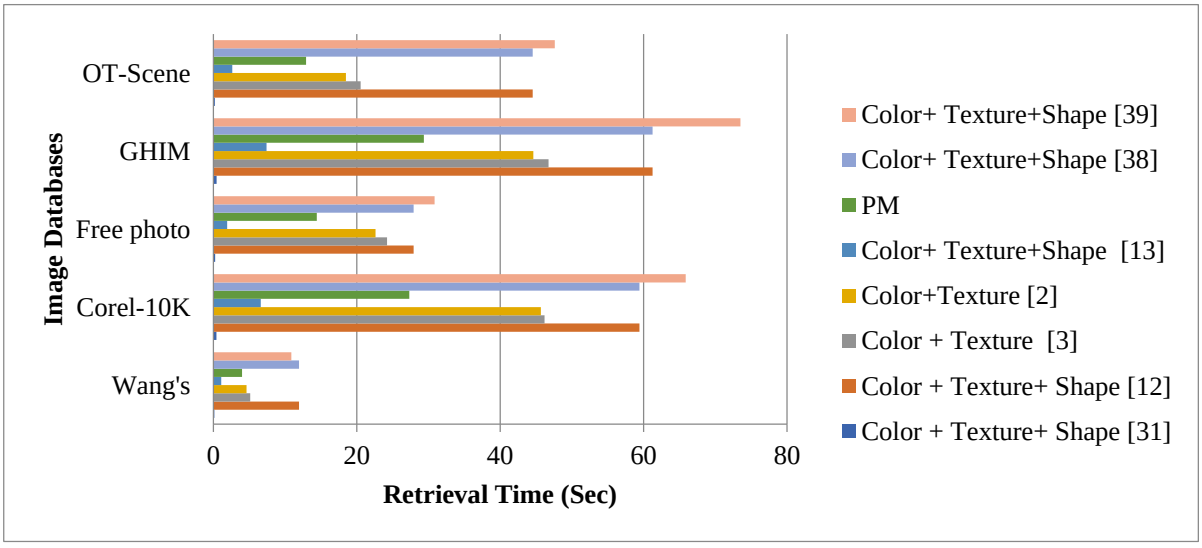

Figure 7. Retrieval time (in seconds) of the existing and proposed feature integrated CBIR systems.

4. Conclusions

The new two-level similar image searching system introduced in the proposed CBIR system used the image selection rule at the first level of the search process. This initial process selects a greater number of similar images to the second searching step. In this second step, the proposed framework has taken the most dominant color and radial difference pattern from the query and selected images. The color and texture features similarity of the query and database images are measured by taking the distance information of them. Then the proposed work assigned the dynamic optimal weight to the color and texture features distance information by using the FFO algorithm. This framework approximately gives 88.98%, 63.18%, 83.25%, 70.68% and 70.31% average retrieval precision over the Wang's, Corel -10K, free photo, GHIM and OT-scene. However, due to the number of particles and iterations of the FFO algorithm it takes high retrieval time for the CBIR system using large databases. The future scope of this work is to explore different kinds of optimization algorithms over the multidimensional feature vectors rather than the two dimensional vectors used in this work and optimal number of dominant color cluster formation over the given image.

Conflict of interest

The authors declare that they have no conflict of interest.

References

1. J. Yue, Z. Li, L. Liu, Z. Fu, Content-based image retrieval using color and texture fused features, *Math. Comput. Model.,* **54** (2011), 1121–1127.
2. M. Singha, K. Hemachandran, A. Paul, Content-based image retrieval using the combination of the fast wavelet transformation and the colour histogram, *IET Image Process.*, **6** (2012), 1221–1226.
3. X. Y. Wang, B. B. Zhang, H. Y. Yang, Content-based image retrieval by integrating color and texture features, *Multimed. Tools. Appl.*, **68** (2014), 545–569.
4. M. Dey, B. Raman, M. Verma, A novel colour- and texture-based image retrieval technique using multi-resolution local extrema peak valley pattern and RGB colour histogram, *Pattern Anal. Appl.*, **19 (**2016), 1159–1179.
5. X. Y. Wang, L. L. Liang, Y. W. Li, H. Y. Yang, Image retrieval based on exponent moments descriptor and localized angular phase histogram, *Multimedia Tools. Appl*., **76** (2017), 7633–7659.
6. G. Sucharitha, R. K. Senapati, Biomedical image retrieval by using local directional edge binary patterns and Zernike moments, *Multimed Tools. Appl*., **79** (2020), 1847–1864.
7. R. P. Y. Narasimha, L. K. Pavithra, T. S. Sharmila, Analysis of Supervised and Unsupervised Learning in Content Based Multimedia Retrieval, *Int. Conf. on Comput. Commun. Sign. Proc.* (ICCCSP), (2018), 1–5.
8. X. Y. Wang, Y. J. Yua, H. Y. Yang, An effective image retrieval scheme using color, texture and shape features, *Comput. Stand. Inter.*, **33** (2011), 59–68.
9. X. Y. Wang, H. Y. Yang, D. M. Li, A new content-based image retrieval technique using color and texture information, *Comput. Electr. Eng.*, **39** (2013), 746–761.

10. A. Khokher, R. Talwar, A fast and effective image retrieval scheme using color-, texture-, and shape-based histograms, *Multimed Tools. Appl*., **76** (2017), 21787–21809.
11. N. Varish, A. K. Pal, R. Hassan, M. K. Hasan, A. Khan, N. Parveen, et al., Image Retrieval Scheme Using Quantized Bins of Color Image Components and Adaptive Tetrolet Transform, *IEEE Access*, **8** (2020), 117639–117665.
12. S. Fadaei, R. Amirfattahi, M. R. Ahmadzadeh, New content-based image retrieval system based on optimised integration of DCD, wavelet and curvelet features, *IET Image Process.*, **11** (2017), 89–98.
13. L. K. Pavithra, T. S. Sharmila, An Efficient Framework for Image Retrieval using Color, Texture and Edge Features, *Comput. Electr. Eng.,* **70** (2018), 580–593.
14. L. Putzu, L. Piras, G. Giacinto, Convolutional neural networks for relevance feedback in content based image retrieval, *Multimed. Tools Appl.*, **79** (2020), 26995–27021.
15. L. Pinjarkar, M. Sharma, S. Selot, Deep CNN combined with relevance feedback for trademark image retrieval. *J Intell Sys*., **29** (2020), 894–909.
16. K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, *Comput. Vis. Pattern Recognit*., (2014), 1–6.
17. X. Li, J. Yang, J. Ma, Large Scale Category-Structured Image Retrieval for Object Identification Through Supervised Learning of CNN and SURF-Based Matching, *IEEE Access*, **8** (2020), 57796–57809.
18. A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, H. Chen, Harris hawks optimization: Algorithm and applications, *Future Gener Comput Syst.*, **97** (2019), 849–872.
19. G. G. Wang, Moth search algorithm: a bio-inspired metaheuristic algorithm for global optimization problems, *Memetic Comp*., **10** (2018), 151–164.
20. Q. Guo, Y. Quan, C. Jiang, Object Pose Estimation in Accommodation Space using an Improved Fruit Fly Optimization Algorithm, *J. Intell. Robot. Syst.*, **95** (2019), 405–417.
21. G. G. Wang, S. Deb, Z. Cui, Monarch butterfly optimization, *Neural Comput & Applic.,* **31** (2019), 1995–2014.
22. S. Li , H. Chen, M. Wang, A. A. Heidari, S. Mirjalili, Slime mould algorithm: A new method for stochastic optimization, *Future Gener. Comput. Syst.*, **111** (2020), 300–323.
23. Y. Yang, H. Chen, A. A. Heidari, A. H. Gandomi, Hunger games search: Visions, conception, implementation, deep analysis, perspectives, and towards performance shifts, *Expert Syst. Appl.,* **177** (2021), 1–34.
24. Y. Fan, P. Wang, M. Mafarja, M. Wang, X. Zhao, H. Chen, A bioinformatic variant fruit fly optimizer for tackling optimization problems, *Knowl. Based Syst.,* **213** (2021), 1–21.
25. H. Yu, W. Li, C. Chen, J. Liang, W. Gui, M. Wang, et al., Dynamic Gaussian bare-bones fruit fly optimizers with abandonment mechanism: method and analysis, *Eng. Comput.*, (2020), 1–29.
26. X. Wang, H. Chen, A. A. Heidari, X. Zhang, J. Xu, Y. Xu, H. Huang, Multi-population following behavior-driven fruit fly optimization: A Markov chain convergence proof and comprehensive analysis, *Knowl. Based Syst.*, **210** (2020), 1–26.
27. Y. Fan, P. Wang, A. A. Heidari, M. Wang, X. Zhao, H. Chen, C. Li, Rationalized fruit fly optimization with sine cosine algorithm: A comprehensive analysis, *Expert Syst. Appl.*, **157** (2020), 1–23.
28. Y. Fan, P. Wang, A. A. Heidari, M. Wang, X. Zhao, H. Chen, et al., Boosted hunting-based fruit fly optimization and advances in real-world problems, *Expert Syst. Appl.*, **159** (2020), 1–23.

29. N. C. Yang, W. H. Chang, C. M. Kuo, T. H. Li, A fast MPEG-7 dominant color extraction with new similarity measure for image retrieval, *J. Vis. Commun. Image. R.*, **19** (2008), 92–105.
30. E. Walia, S. Vesal, A. Pal, An Effective and Fast Hybrid Framework for Color Image Retrieval, *Sens. Imaging*, **15** (2015), 1–23.
31. S. M. Lin, Analysis of service satisfaction in web auction logistics service using a combination of Fruit fly optimization algorithm and general regression neural network, *Neural. Comput. Appl.*, **22** (2013), 783–791.
32. E. Walia, A. Pal, Fusion framework for effective color image retrieval, *J. Vis. Commun. Image R.,* **25** (2014), 1335–1348.
33. Y. Li, M. Han, Improved fruit fly algorithm on structural optimization, *Brain Inf.*, **7** (2020), 1–13.
34. X. Li, J. Yang, J. Ma, Recent developments of content-based image retrieval (CBIR), *Neurocomputing*, **452** (2021), 675–689.
35. Y. Yang, H. Chen, S. Li, A. A. Heidari, M. Wang, Orthogonal learning harmonizing mutation-based fruit fly-inspired optimizers, *Appl. Math. Model.*, **86** (2020), 368–383.
36. G. Ding, Y. Qiao, W. Yi, W. Fang, L. Du, Fruit fly optimization algorithm based on a novel fluctuation model and its application in band selection for hyperspectral image, *J. Ambient Intell. Human Comput*., **12** (2021), 1517–1539.
37. X. Zhang, Y. Xu, C. Yu, A. A. Heidari, S. Li, H. Chen, et al., Gaussian mutational chaotic fruit fly-built optimization and feature selection, *Expert Syst. Appl.,* **141** (2020), 1–14.
38. S. Singh, S. Batra, An efficient bi-layer content based image retrieval system. *Multimed Tools Appl.*, **79** (2020), 17731–17759.
39. U. A. Khan, A. Javed, R. Ashraf, An effective hybrid framework for content based image retrieval (CBIR), *Multimed Tools Appl.,* **80** (2021), 26911–26937.

AIMS Press