Chapter 3: Design and Implementation of Effective CBIR System Using Fusion of Features and Machine Learning Approaches

In this chapter, design and implementation of two frameworks for effective CBIR system are presented, which are based on the different variants of computationally light weighted colour and texture features, weighted similarity measures and machine learning approaches. This chapter is organized in two-fold viz. in first fold, it presents improved CBIR system using a fusion of fast features with varying weighted similarity measure and random forests. This section includes the fusion of chromaticity moments, colour percentiles and LBP features with varying weighted similarity measure for image representation and searching. Further, in another framework random forests is used on the pre- manual clusters of images based on the known ground truth of all images.

In second fold, it presents fast and effective image retrieval based on supervised learning framework with fusion of orthogonal combination of local binary patterns (OCLBP) and statistical moments. In this work, a supervised learning based image management is used as a prior step for speeding up image retrieval in the large database. Basically, this framework automatically clustered the images based on some known ground truth from each class of images, and then go for the retrieval.

Introduction of both contributions presented in this chapter are given in section 3.1. In section 3.2 of this chapter, an improved CBIR system using fusion of fast features with varying weighted similarity measure and random forests is presented. Further, in section 3.3 of this chapter presents a fast and effective CBIR system based on supervised learning with combination of OCLBP and statistical moments.

3.1 Introduction

CBIR is required because most of the system depend on image search engines rely purely on the textual descriptor, and this creates a lot of false detection in the outcomes. Content based recovery of images depends on the features of images. For the description of images, CBIR methods generally used either colour, texture or shape based features [60-86]. In feature extraction methods, local patterns with spectral content have made their place because of their efficiency and simplicity. Usually colour, texture, and shape are the basic discriminatory properties for any image.

In the beginning of research related to CBIR systems, colour-texture, and shape features were used independently, but in current era various methods have jointly merged all these features [9, 70-86], and used some direct similarity measures for image retrieval. This combination gives encouraging retrieval results by resolving the shortcomings of both. As an individual colour features cannot distinguish between similar images of varying illumination, while texture features can. Contrarily, texture features are dependent on the spatial layout, while colour features are not. Hence, a combination of colour and texture features resolves the shortcomings of both. Feature vector representation and similarity measurement are very critical for retrieval performance of a CBIR system. So, it is very important to extract a fast, informative and accurate feature vector from each image. Therefore, to reduce the retrieval gap and fast feature indexing, this chapter proposes fast hybrid features by combining the chromaticity moments-colour percentile based colour features with LBP based texture feature. Here, chromaticity moments capture the spectral description an image, while colour percentile hold the colour intensity information at a particular level, also very fast in indexing and quite independent of colour space. For texture feature, we have

used local binary pattern, derived from the local neighbourhood of each pixel in the image. The advantages of this fused features sets are: fast in feature indexing, isolation from varying image size, compactness, and its performances. Further, for calculating the similarity measure a new concept of inverse variance weighted Euclidean distance is incorporated. During the calculation of similarity measure, it calculates the weight according to the distribution of features and assigns the priority (weight) to each feature. These priority weights reduce the effect of redundancy and effectively retrieve relevant images.

In recent years, varieties of retrieval methods have been planned but even then issues like semantic gap and searching time exist. Such problems poses fundamental challenge of artificial intelligence (AI) from a high-level perspective that is how to build and train to intelligent machines like human to challenge practical tasks. As reported in Chapter 2 in literature review section that conventional CBIR techniques simply extract low-level features from images and use direct rigid functions to calculate the distance for matching of all images. However, these methods use exhaustive linear search, and are not able to capture the high-level human understanding of similar visual content. It is very important to narrow down the search space and increase the retrieval accuracy. So for improving the semantic gap and response time, machine learning and deep learning [11] can be promising techniques that attempts to address this challenge in the long-term. But deep learning required huge computational time for the training of the model. Therefore, we have used machine learning based technique where it utilizes supervised learning approach as prior step for speeding up retrieval performance in the large database. This work proposed additional content-based query image classification cum retrieval model on the pre-assumed clusters. In this contribution, images are assumed in the given cluster as per the database details, and supervised query image

classification and retrieval model filters out all irrelevant images and retrieve most relevant images in less search time. As, we know that for successful usage in query image annotation and retrieval, the generalization capabilities of classifiers are most essential [132]. Therefore, to overcome the over-fitting problem we have introduced random forests classifier. The random forests [133] is an ensemble approach based on divide-and-conquer approach. The basic concept behind ensemble method is that a group (forest) of classification tree can come together to form a strong learner, and final classification of an individual is determined by voting over all trees in the forest. So this final prediction based on combining the prediction of many tress increases the confidence of results. Both contributions are discussed in section 3.2, where at first we demonstrate about proposed framework, start with feature extraction, followed by retrieval using variance based similarity measure and classification framework.

Since the above contribution using random forests is based on pre-assumed clusters of images, in another framework, at first we have clustered the images based on the some known ground truth and then further go for the retrieval. In this contribution, light weighted colour moments and orthogonal combination of LBP features are used as colour and texture features respectively. Since long histogram in local binary patterns increases the feature dimension, storage space and similarity calculation therefore for capturing the texture characteristics, we use the orthogonal combination of LBP [135]. We have incorporated two extra statistical features (i.e. Kurtosis and Entropy) with colour moments. Using these features, we have proposed another content-based image classification-cum-retrieval model which is treated as a pre-processing step, used for image grouping and retrieval based on small set of known ground truth. In this contribution, we concentrate on image databases and hence tackle the problem of developing fast and efficient ways for image retrieval. As we already reported that

general CBIR techniques use exhaustive linear search, and are not able to capture the high-level human understanding of similar visual content. In this contribution, we try to bridge this semantic gap by applying machine learning techniques before matching images based on their features. In proposed method, we first pre-classified the images using support vector machines (SVM) and then retrieve similar images from a pre-classified image dataset. SVM is a machine learning technique for classifying both linear and non-linear data [136]. In the case of linearly separable data, SVM searches for the best hyperplane that separates the data with the largest margin. The margin is the shortest distance between the training tuples on one side of the hyperplane and the hyperplane itself such that it is equal to the shortest distance of the hyperplane with the training tuples on the other side of the hyperplane. Tuples (image samples) that lie on this shortest distance are known as support vectors. For the case of linearly inseparable data, the data is transformed into a higher dimensional space using a non-linear function. During the training of SVM, the process starts with the choice of a kernel function through which patterns are mapped into a higher-dimensional space. According to Hsu et al. [137], a radial basis function (RBF) kernel is a reasonable choice in most cases, particularly for a small number of features. In this study, we have to deal with relatively few features, so we have taken RBF kernel function.

For retrieval, the trained supervised model is treated as the pre-processing step. Further, a prior computing similarity measure is applied to speed up the retrieval performances and searching time. Experimental results show that our method performs better than the conventional framework of CBIR. Moreover, it is much more efficient than the traditional methods due to the minimized search domain in our approach. Thus, we are able to efficiently influence semantic estimation. The organisation of the rest of this contribution is given in section 3.3.

3.2 Improved CBIR System Using Fusion of Fast Features with Varying Weighted Similarity Measure and Random Forests

This work introduces an efficient and fast CBIR system, which is based on the combination of computationally light weighted colour and texture features viz. chromaticity moments, colour percentiles, and local binary patterns. For searching, this work proposes inverse variance based varying weighted similarity measure (low for high variance feature and high for low variance feature), which reduces the effect of redundancy by assigning the priority to each feature, and effectively retrieves relevant images. In addition, this chapter also proposes query image classification and retrieval model by filtering out irrelevant class images using random forests (RF) classifier, which recovers the class of a query image based on distinct learning (supervised) of various decision trees. This successful ensemble classification of query images reduces the semantic gap, searching space, and enhances the retrieval performance. Extensive experimental analyses on benchmark databases confirm the usefulness and effectiveness of this work.

3.2.1 Methods and Models

In this section, first we explore the proposed work based on fusion of colour and texture features using inverse variance weighted similarity measure. Further, we introduce the query image classification and retrieval model. For the implementation of this model, it is highly desirable to make groups of meaningful categories. Here in this work, we pre-assumed the group of the images according to the description given in database, and apply the concept of image classification, where an image is classified according to its visual content.

This work presented in this chapter is divided into 3 sections (3.2.1.1, 3.2.1.2, and 3.2.1.3), where section 3.2.1.1 gives the details of extraction procedure for used features. Sections 3.2.1.2 and 3.2.1.3 give two different proposed methods for image retrieval. Section 3.2.1.2 demonstrates image retrieval using varying weighted similarity measures between query image features and other database images features. Section 3.2.1.3 gives the details of image classification and retrieval model using random forests. The outcomes of both methods are significantly better in terms of accuracy and other issues, which are discussed in section 3.4.1.

3.2.1.1 Feature Extraction

This work introduces chromaticity moments-colour percentile based colour features and LBP based texture feature, very fast in indexing and gives better retrieval performance. The details of chromaticity moment and other features are discussed in next given sections.

3.2.1.1.1 Chromaticity Moments

Chromaticity moments are statistics, based on the concept of the chromaticity diagram, defined within the CIE xyY colour space.

To define the concept of chromaticity diagram in xyY colour space, Firstly we transform it in XYZ colour space by using nonlinear transformation of the RGB colour space [35].

$$X = 0.607 * R + 0.174 * G + 0.200 * B \quad (3.1)$$

$$Y = 0.299 * R + 0.587 * G + 0.114 * B \quad (3.2)$$

$$Z = 0.066 * R + 1.111 * B \quad (3.3)$$

Then (x, y) chromaticity are defined as:

$$x = \frac{X}{X+Y+Z} \quad (3.4)$$

$$y = \frac{Y}{X+Y+Z}$$

(3.5)

For given x, y ϵ [0, 1], we quantize x and y to chosen levels, X_q and Y_q, respectively. So this 2D representation for an image where, each pixel (from RGB triplet) holds a pair of (x, y) chromaticities are known as chromaticity diagram. Further, we can calculate the trace of its chromaticity set as:

$$Shape(x,y) = \{1 \text{ if } \exists\ (i\ ,j)\text{: I(i, j) } Yields(x,\ y)$$

$$0 \qquad \text{otherwise }\},\ 0 \le i < M-1, 0 \le j < N-1$$

where I is an image of dimensions M×N

In this equality, we are setting an entry is 1 if the corresponding probability density is positive, 0 otherwise

2-Dimensional distributions is calculated for more than one pixels holding the same (x, y) pair using

Distribution (x, y) = #pixels Yielding(x, y).

Here, Distribution (x, y) is the estimated as bi-dimensional probability density function in the xy plane of an image, and Shape(x, y) is its trace.

Finally, we calculated Shape and Distribution as a set of chromaticity moments, defined, respectively, as:

$$M_{Shape}(a\ ,b\ ) = \sum_{x=0}^{Xq-1}\sum_{y=0}^{Yq-1} x^{\,a}y^{b}\ Shape(x,y) \qquad (3.6)$$

$$M_{Distribution}(a\ ,b) = \sum_{x=0}^{Xq-1}\sum_{y=0}^{Yq-1} x^{\,a}y^{b} Distribution(x,y) \qquad (3.7)$$

where a = 0; 1; 2; . . . , b = 0; 1; 2; . . ., and X_q, Y_q are the dimensions of the xy space. Means, X_q=Y_q being the number of quantization levels for each of the two

chromaticity histogram dimensions x; y. M_{shape} and $M_{Distribution}$ form the set of chromaticity moments of image I. This set of features is extracted for each image in the database, and stored along with the images. In this experiment, five Trace type and five Distribution type moments are extracted using the following values of the exponents and quantization levels: (a, b) ϵ {(0, 1); (1, 0); (1, 1); (1, 2); (2, 1)}, and $X_q=Y_q=100$.

Procedure for chromaticity moment features extraction:

1. Conversion an RGB space image into xyY plane
    1.1.Load an image
    1.2. Convert image into XYZ plane, such as I_{XYZ}
    1.3.Covert I_{XYZ} into xyZ plane.
2. ComputeChromaticityMomentfeatures ($I_{xyY,}$ exponent)
    2.1. Compute the trace and distribution of xyZ plane images
    2.2. Compute the chromaticity moments (M_{shape} and $M_{Disribution}$) using Eq. (3.6) and (3.7).
    2.3 Normalize the moments features by dividing it from the corresponding maximum attainable value

3.2.1.1.2 Colour Percentiles

Colour percentile is very simple, quite independent of colour space and computationally light weighted features, used for visual retrieval and inspection [130-131]. The colour features used in this study are based on percentiles of vectorized one dimensional *R, G*, and *B* colour channels. Colour percentile for an image of matrix *Xi*, with percentages *Pr* in the interval [0 to 100] is depends upon nature of data. If *Xi* is a vector, then percentile *Pr* is a vector or a scalar with the same length, means *Y(i)* contains the *Pr(i*) percentile. Percentile uses linear polynomials to find y_i = f(x_i), the values of the

underlying function $Y = f(X)$ at the points in the vector x. Given the data points (x_1, y_1) and (x_2, y_2), where $y_1 = f(x_1)$ and $y_2 = f(x_2)$, linear interpolation finds $y = f(x)$ for a given x between x_1 and x_2 as follows [134]:

$$y = y_1 + \frac{x - x_1}{x_2 - x_1}(y_2 - y_1) \tag{3.8}$$

Here x_1 and x_2 are minimum and maximum value of input data, and y_1 and y_2 are range interval [0,100]. Percentile values for this interval may be unreliable because of saturated noise in the images, and it is safer to use the intermediate range [131]. Therefore, in this study, percentile intervals are randomly set as 25, 50 and 75, which covers the approx. minimum to approx. maximum colour range with minimum noise effect.

Procedure for colour percentile feature extraction:

1. Load an image
2. Calculate the dimension of the image viz. number of rows (R), columns (C) and channels.
3. Input the percentile intervals 25, 50, and 75.
4. Vectorized colour pixels intensity of an image, viz. R×C pixel representation of an image converted into R*C×1 representation.
5. After vectorization of an image pixels values, calculate the percentile features vector for given intervals
6. Repeat the step-5 for all channels.
7. Concatenate all channel features, and normalize it using min-max normalization.

3.2.1.1.3 Local Binary Patterns

LBP is a computationally light weighted texture features, derived from the local neighbourhood of each pixel in the image [42]. This operator treats each pixel as a

center pixel, and calculates the difference of center pixel with neighbourhood pixel, and multiplies it with a binary image window. In this work, LBP features are extracted from luminance plane. LBP operator is mathematically defined as:

$$LBP_{P,R} = \sum_{i=0}^{P-1} 2^i D_1(g_i - g_c) \tag{3.9}$$

where $$D_1(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases}$$

The gray scale values of neighborhood pixel and center pixel is denoted as g_i and g_c. The connectivity from neighbouring pixels is represented as P and the neighborhood radius is denoted as R. Finally, for an image of size M*N with L intensity value, histogram of LBP image is represented as texture features, denoted as Hist(L)|LBP and defined in Eq.(3.10).

$$Hist(L)|LBP = \sum_{x1}^{M} \sum_{x2}^{N} D_2(LBP(x_1, x_2), L) \tag{3.10}$$

where $$L = [0,\ 2^{P-1}]$$

$$D_2(x_1, x_2) = \begin{cases} 1 & x_1 = x_2 \\ 0 & otherwise \end{cases}$$

Procedure for LBP features extraction:

1. Load an image.
2. Convert into gray scale image
3. For each pixel in window, compare it with P=8 and R=1 neighbouring pixels.
4. Follow the pixel in clockwise or anti clock wise direction and complete a circle
5. If window underneath centre pixel is greater than neighbour pixel, then use 1 else use 0.

6. After iii step, we get 8 bit binary sequence number, compute the histogram over the window of the frequency of each number occurring using Eq.(3.10)
7. Normalize the histogram by dividing the total number of pixels.
8. Concatenate normalized histogram, this is final feature vector.

3.2.1.2 Image Retrieval Using Fusion of Features and Weighted Similarity Measure

To retrieve images based on fusion of different feature sets and inverse variance weighted similarity measure, the proposed CBIR systems are divided into two sections, working diagram is shown in Fig.3.1. In the first section, Chromaticity moments, Colour Percentile, and LBP features are extracted from each image in the database, and after fusion these features are used to index the images; finally they are stored into the database along with the images.

Let CrM, CPr, and LBP are feature sets of chromaticity moments, colour percentile and local binary patterns, respectively, which are extracted separately and represented as:

$$CrM=[CrM_1\ CrM_2\ \ CrM_3\ \ldots\ldots\ldots\ldots CrM_{10}],$$

$$CPr=[CPr_{11}\ CPr_{12}\ \ CPr_{13}\ldots\ldots\ldots\ldots\ldots\ldots CPr_{19}],\text{ and}$$

$$LBP=[LBP_{20}\ LBP_{21}\ \ LBP_{22}\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots LBP_{55}]$$

Final feature set (FS) are formed after fusion of all these feature sets

$$FS=[CrM \cup CPr \cup LBP]$$

Final feature set for one image = $[CrM_1\ CrM_2\ldots\ldots CrM_{10}\ CPr_{11}\ CPr_{12}\ldots\ldots\ldots CPr_{19}\ LBP_{20}\ LBP_{21}\ldots\ldots\ldots\ldots LBP_{55}]$

If there are *K* number of images then the size of feature matrix=K×55

So, we can write this feature vector into K x n representation, where K is the number of images and n is the number of features, and a feature vector can be expressed in a matrix such as:

$$FS=[FS_{1,i}\, FS_{2,i}.....FS_{J,i}\ldots\ldots\ldots\ldots FS_{K,i}] \text{ , where } i=1 \text{ to } n.$$

Finally, for indexing these entire feature vectors are stored into the database along with the images. In the second section, at a query time, a same feature vector is extracted from the query image, and it is matched using proposed similarity measure against the feature vectors in the database to retrieve closest possible similar images available in the database. For calculating similarity measure, here a new concept of inverse variance weighted Euclidean distance is incorporated, which is described as follows:

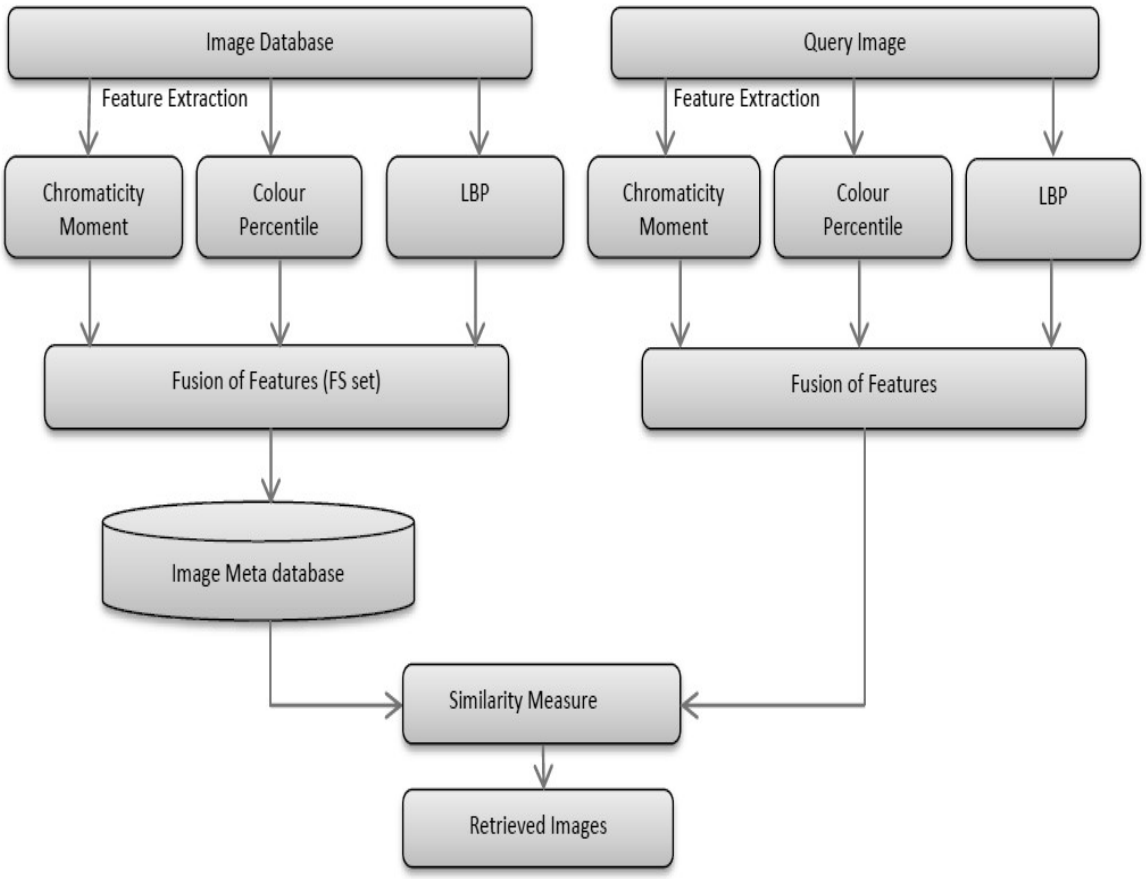

Fig. 3.1: Proposed retrieval framework based on fusion of fast features and weighted similarity measure

This distance metric between two n[th] dimensional vectors can be calculated as:

$$D_{FS,\ Q} = \sqrt{\sum_{i=1}^{n}\left(\frac{FS_i}{\sigma_i} - \frac{Q_i}{\sigma_i}\right)^2} \tag{3.11}$$

where σ_i is standard deviation of the i[th] feature

$$D_{FS,\ Q} = \sqrt{\sum_{i=1}^{n}\frac{(FS_i - Q_i)^2}{\sigma_i^2}} \tag{3.12}$$

$$D_{FS,Q} = \sqrt{\sum_{i=1}^{n} w_i (FS_i - Q_i)^2} \tag{3.13}$$

where $$w_i = \frac{1}{\sigma_i^2} \tag{3.14}$$

FS_i is i[th] feature vector, Q_i is query image feature vector and w_i is weight of i[th] feature. This weight w_i is attached to the i^{th} variable. Here, we just compute the usual squared differences between the original scale variables, and multiply these squared differences by their corresponding weights. These proposed weights are varying in nature (weight of a variable with low variance is high, while the weight of a variable with high variance is low). This interesting property of the varying weights, yield another way of compensatory effect, which effectively retrieved relevant images.

Using Eq. (3.13) similarity measures are calculated for all features, after the calculation, measures are sorted in increasing order and finally top most similar images are retrieved.

3.2.1.2 Image Retrieval Using Random Forests

This framework is based on the detection of query images because correct detection of query images greatly enhances the performance of retrieval by filtering out irrelevant images during matching. In this work, we try to bridge the semantic gap between low-

level feature and high level understanding by applying machine learning techniques. Therefore, to achieve the goal of automatic classification of the query, we have proposed supervised learning based model that capture extracted FS feature set of images, and trained the classifier with corresponding classes. This process is treated as a pre-filtering step and applied prior computing similarity measure to speed up the searching.

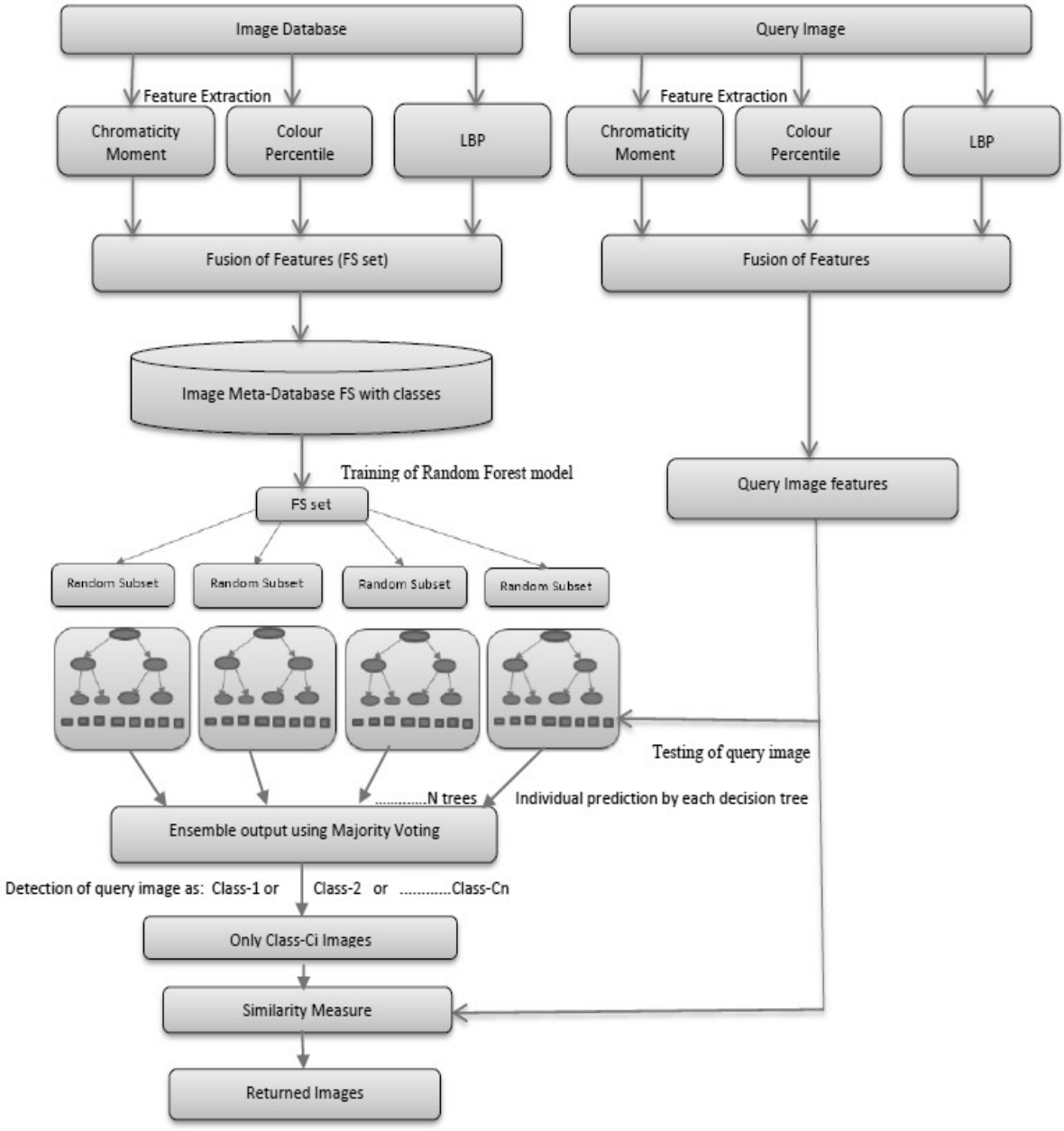

Fig.3.2: Proposed CBIR system based on query classification using random forests

Basically, this modelled classifier works as a detector, who predicts the classes of unknown query images. Fig.3.2 shows the working of proposed CBIR systems, in which first we extract chromaticity moments, colour percentile and LBP features, and then concatenate them with the same weight. Finally, feature vector set (FS) are stored into the database, used by the random forest (RF) classifier for image classification. The basic concepts of random forest are based on the formation of weak decision trees in parallel, further we combine the trees to form a single, strong learner by averaging or taking the majority vote. For the classification of images, feature set (FS) are randomly divided into various subset for the training of decision tree individually, where each node chooses small subset of features at random, for finding a feature which optimized the split. After training of each decision tree, at the query time, same features are automatically extracted from the query image**,** and fed into all trained decision trees. Majority voting based ensemble method which combined the output of all the decision tress, are used for prediction of query image, and searching is automatically narrowed down in the detected class images. Complete algorithm for the modelling and prediction of random forest classification are given below.

Algorithm 3.1 Random Forests

Input: A training set *S (Subset* of FS*) with n* features, and *B number of* trees in

forest.

Steps:

1. function RF(S , n)
2. $X \leftarrow \varphi$
3. $for\ i \leftarrow 1$ to B do
4. $S^{(i)} \leftarrow$ A bootstrap sample from S

5. $x_i \leftarrow$ RandomizedLearnTree $(S^{(i)}, n)$
6. $X \leftarrow X \cup \{x_i\}$
7. end for
8. return X
9. end function

Function *RandomizedLearnTree* $(S^{(i)}, n)$

1. At each node
2. $f \leftarrow$ very small subset of n
3. Split on best feature in f
4. return learned tree
5. end function

Output:

Let $C_b(x)$ is the class prediction of b^{th} random forest tree then final ensemble output random forest.

$$C^B(x) = \text{majority } vote \ \{ C_b(x) \ \}_1^B$$

At the time of similarity matching, searching is done only in the detected class images, show retrieval performances is directly proportional to the Random forest classification performances. So after detecting the class of query image, we can retrieve all the relevant images from database in less searching time. For retrieving more similar images among the filtered images, we have used the weighted ED similarity measure.

3.2.2 Results Analysis and Discussions

The retrieval performance of this work has been analysed in two sections. First section gives the details of results analysis using proposed fusion and inverse variance weighted

similarity measure, and second section gives the descriptions of result analysis using proposed classification framework.

3.2.2.1 Result Analysis for Fusion of Features and Weighted Similarity Measure

- **Wang Database Analysis**

Fig .3.3(a)

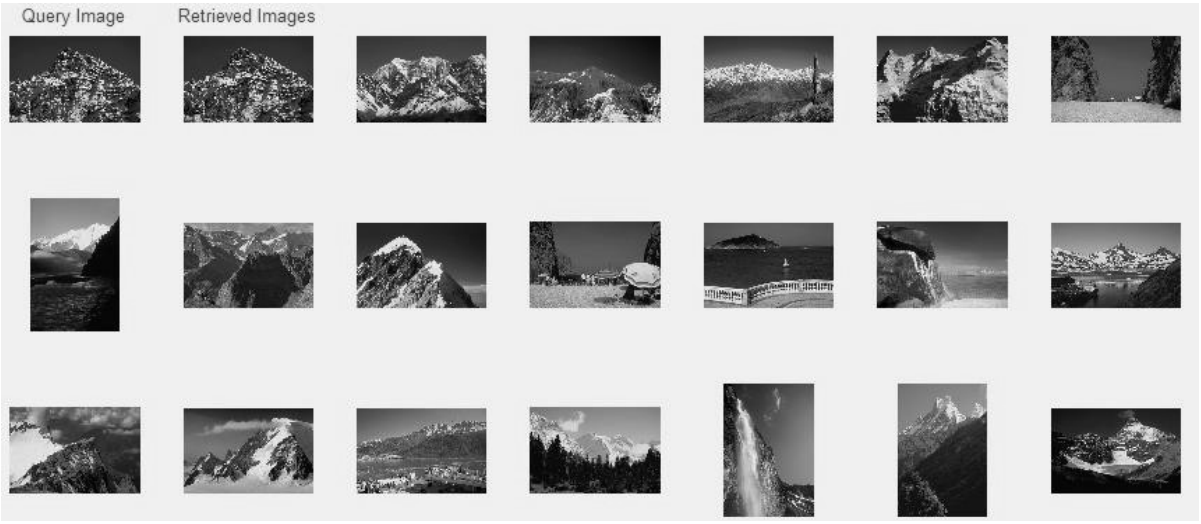

Fig .3.3(b)

Fig. 3.3: Retrieval results using proposed fusion based feature and weighted similarity measure on different sample queries. a. Beach b. Mountain-Hill

Fig.3.3 (a-b) show the snapshot of image retrieval for two different sample queries for Beach, and Mountain class images, where retrieval performances are 90 % for the Beach class and 80 % for the Mountain class. It shows the glimpse of

effectiveness for the proposed framework because both class images are very similar to each other and challenging techniques are required for the correct retrieval.

Table 3.1: Average precision of features using weighted similarity measure

Classes	Chromaticity Moments	Colour Percentile	Local Binary Patterns	Proposed Hybrid Descriptor
Africa	0.60	0.58	0.47	0.67
Beach	0.40	0.56	0.41	0.62
Building	0.26	0.25	0.56	0.61
Bus	0.70	0.48	1.00	1.00
Dinosaurs	0.81	1.00	1.00	1.00
Elephant	0.33	0.60	0.41	0.69
Flowers	0.26	0.50	0.94	0.98
Horses	0.74	0.79	0.52	0.72
Mountain	0.65	0.43	0.28	0.70
Food	0.65	0.71	0.47	0.72
Average	**0.540**	**0.590**	**0.606**	**0.771**

For the descriptive analysis, the experiments are carried out to randomly select 10 % of each category as the query images with the number of retrieved images set as 20 for different comparative discussions. In Table 3.1, we have shown the CBIR performance in term of average precision as individual features and with combination, whereas individual LBP and Colour percentile performance are nearly same, 59 % and 60.6 %, respectively, and chromaticity moment gives 54 %, while proposed fused features gives 77.1 % with 100% retrieval performance for the classes of Bus, Dinosaur and 98 % for Flowers. In this work, minimum retrieval performances are 61 % and 62 % for the Building and Beach classes respectively because both classes have wide variety of patterns. For the query of Beach class images, except those truly containing a Beach, maximum other retrieved images belong to the Mountain class, which shows there is some sort of similarity between both the classes.

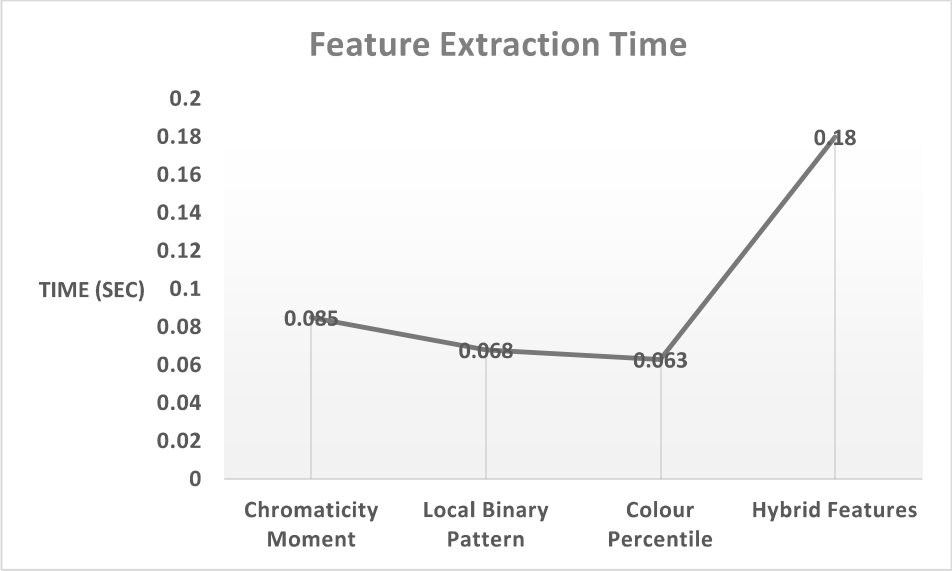

Fig. 3.4: Average feature extraction time for one image

Fig.3.4 shows the required time for feature extraction. For an image, it takes an average time 0.068 second for LBP, 0.085 second for Chromaticity moments, 0.063 second for colour percentile, and 0.180 second for hybrid features. From all these analysis it is clear that proposed feature extraction is very fast and newly incorporated colour percentile feature is faster than all. Also, due to moderate dimension (only 55 features) of proposed feature set, it reduces the calculation of similarity measure to some extent**.**

Table 3.2 and Fig. 3.5 show the comparative class-wise average precision, overall average precision (OAP) and recall for all 10 classes. In Table 3.2 as per the class-wise analysis our methods is lagging from some researcher for a particular class but at the same time our retrieval performances are significantly encouraging from the same researcher for different classes . For each class our retrieval performances are not as degraded as other methods. Other methods retrieval performances are encouraging for some classes, and degraded in another class. There is not a single method whose retrieval performance is better than proposed work for three classes. For more

clarification, we can analyse minimum average precision, maximum average precision and overall average precision (mean of classes). From the Table 3.2, we can see that our minimum retrieval performance is significantly encouraging than all methods. It means, our retrieval performances are not as degraded as other methods. Also, maximum achieved average precision is better than all methods. Further, overall mean of average precision confirms the effectiveness of this work, which is encouraging than all. From this Table, it is also clear that our retrieval performances are more generalized than other methods because other methods performances are quite good for some classes and worse for some other classes.

Table 3.2: Class-wise comparative analysis of average precision

Methods	Classes										
	Africa	Beach	Building	Bus	Dinosaur	Elephant	Flowers	Horses	Mountain	Food	**OAP**
Walia and Pal [9]	0.50	0.88	0.56	0.73	1.00	0.84	0.99	1.00	0.77	0.37	0.764
Liu *et al.* [33]	0.54	0.51	0.38	0.46	1.00	0.63	0.90	0.93	0.48	0.39	0.622
PS and Pujari [59]	0.48	0.34	0.36	0.61	0.95	0.48	0.61	0.74	0.42	0.50	0.549
Yue *et al.* [60]	0.59	0.41	0.42	0.72	0.75	0.65	0.83	0.69	0.45	0.45	0.596
Jalab [61]	0.32	0.61	0.39	0.40	1.00	0.56	0.89	0.65	0.56	0.44	0.582
Shenet *et al.* [62]	0.68	0.65	0.52	0.97	0.99	0.52	0.83	0.78	0.61	0.73	0.728
Irtaza *et al.* [63]	0.65	0.60	0.62	0.85	0.93	0.65	0.94	0.77	0.73	0.81	0.755
Rahimi *et al.* [64]	0.42	0.45	0.49	0.31	0.55	0.70	0.61	0.57	0.44	0.44	0.498
Zhao et al. [67]	0.55	0.57	0.57	0.81	0.89	0.73	0.81	0.67	0.49	0.53	0.662
Mistry et al. [68]	0.62	0.60	0.52	0.89	0.94	0.61	0.89	0.63	0.63	0.59	0.696
Proposed	0.67	0.62	0.61	1.00	1.00	0.69	0.98	0.72	0.70	0.72	**0.771**

There is no consistency in retrieval performance for all class images. Hence, these methods cover limited number of aspects and face problem of generalizations. But proposed work retrieval performance is more consistent and encouraging for all classes.

So, these results are encouraging as compare to other state-of–art methods, also due to fast feature extraction and small dimension, it required fewer amounts of indexing time and storage space. The outcomes of proposed work in terms of overall average precision and recall are 77.1 % and 15.42%, respectively.

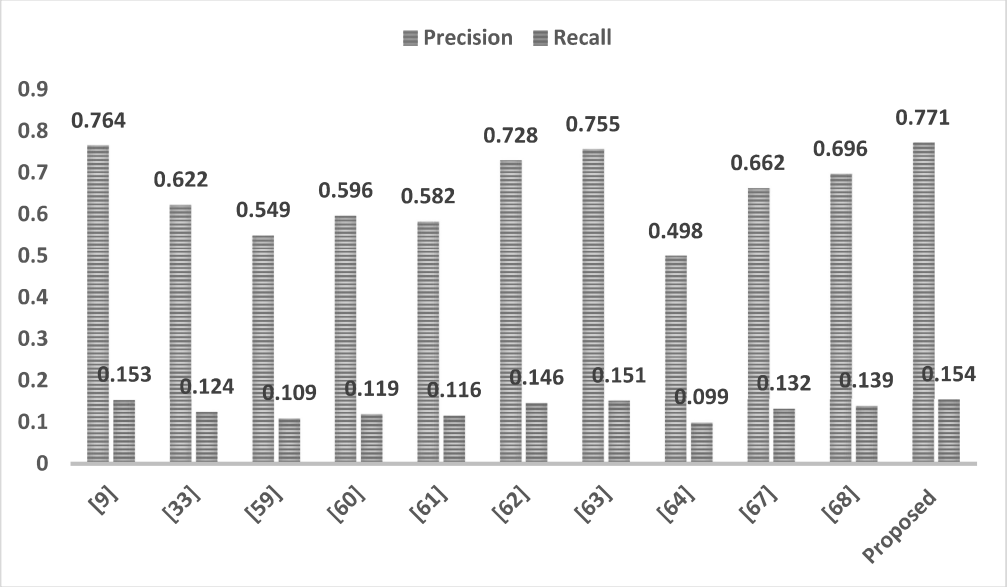

Fig. 3.5: Comparative analysis of the proposed work with other methods

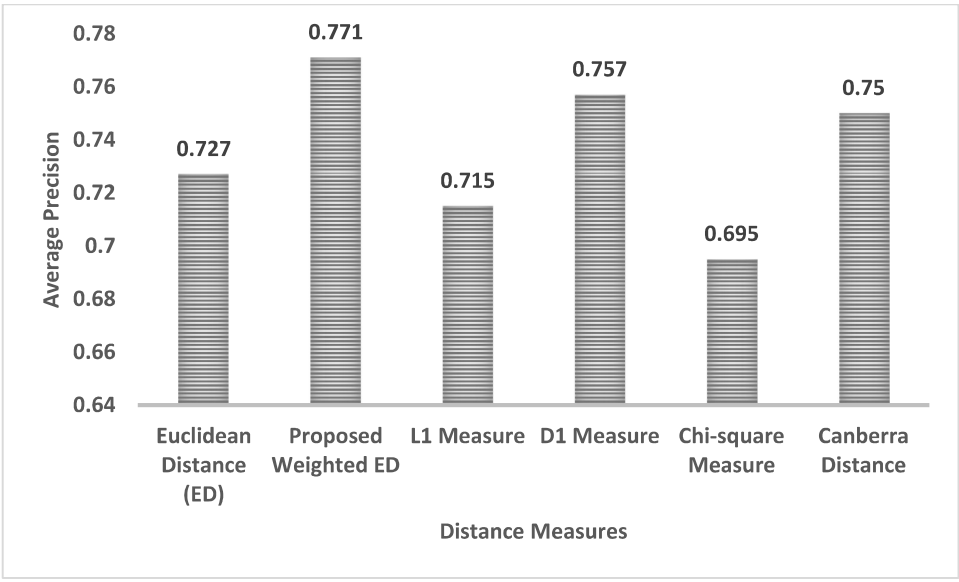

Fig. 3.6: Overall average precision compared with different measures

Here, we have also compared the performance in terms of average precision with Euclidean distance (ED) and other measures. From the Fig. 3.6, it is reflected that the performance of proposed weighted ED is better than ED, L1, D1, Chi-square, and Canberra measures. The overall average precision of proposed weighted measure is better by 4.4 %, 5.6%, 1.4%, 7.6% and 2.1 % with respect to overall average precision of approach introduced by ED, L1, D1, Chi-square and Canberra measures respectively.

Further, retrieval performance of this work is also tested on Corel-10000 database where 10 images are used as query from each class and 20 images are set to retrieve. The overall average precision of Chromaticity moments, colour percentile and local binary patterns are 39.34 %, 41.36 and 46.24% respectively. The proposed fusion framework increases the overall average precision to 63.187 %, which is significantly encouraging than individual features.

- **OT Scene Database Analysis**

For the analysis of OT Scene database, the experiments are carried out, to randomly select 10 images from each category as the query images with the number of retrieved images set as 20 for different comparative discussions. Fig.3.7 shows the snapshot of image retrieval for two sample queries, where retrieval for Forest and Street class images are shown in Fig. 3.7(a), 3.7 (b), respectively, and it gives the glimpse of effectiveness for the proposed work, where retrieval accuracy is 100 %.

In Fig.3.8, we have shown the comparative analysis of CBIR performances in terms of average precision, where this work gives encouraging average precision rate, 65.66 %. The obtained average precision of Forest, Highway, Mountain, and Street classes are 80%, 77%, 73%, and 80%, respectively. With comparison to other state-of-art methods, this work retrieval performance is consistently better for almost all classes.

The obtained overall average precision is significantly encouraging than other state-of-art methods.

Fig. 3.7 (a)

Fig.3.7 (b)

Fig. 3.7 (a-b): Retrieval results for two sample queries: (a). Forest and (b). Street for OT scene database.

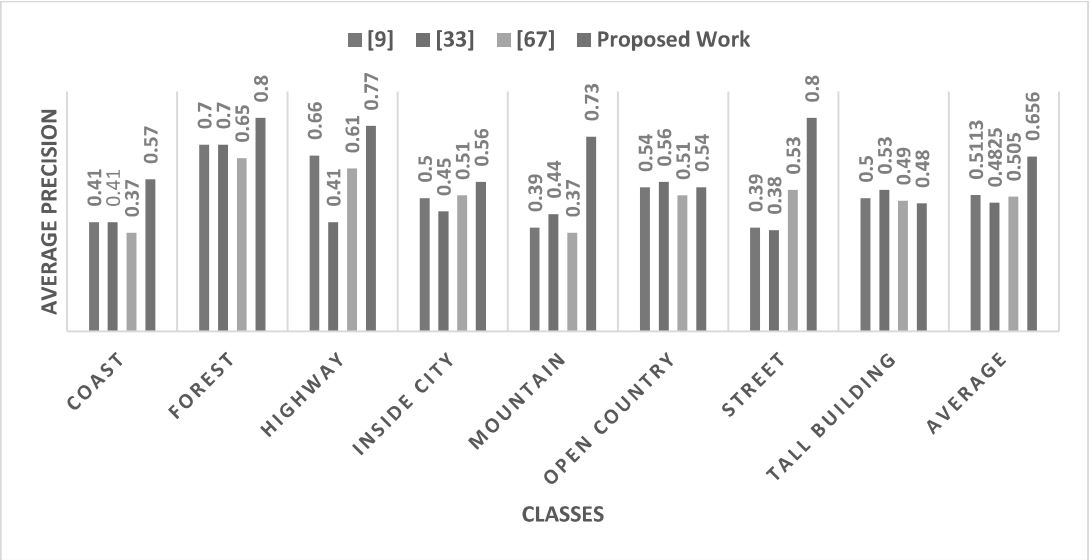

Fig.3.8: Comparative analysis of overall average precision

3.2.2.2 Result Analysis of Image Retrieval for Random Forests Framework

For other proposed classification framework, retrieval performances are evaluated on the basis of classifier performance because the success of filtration of irrelevant images is totally depends upon the detection of relevancy for query image. Here, the proposed framework effectively used for image retrieval, and have given encouraging results. The main reason behind this improvement is exact recognition of query image and filtrations of irrelevant images during matching. Since, we do the distance calculations only in the class to which our query image belongs, show that the performance of image retrieval is directly equal to the classifier performance. Descriptive analysis of this work for Wang and OT scene database are given in the next sections.

- **Wang Database Analysis**

Fig.3.9 (a)

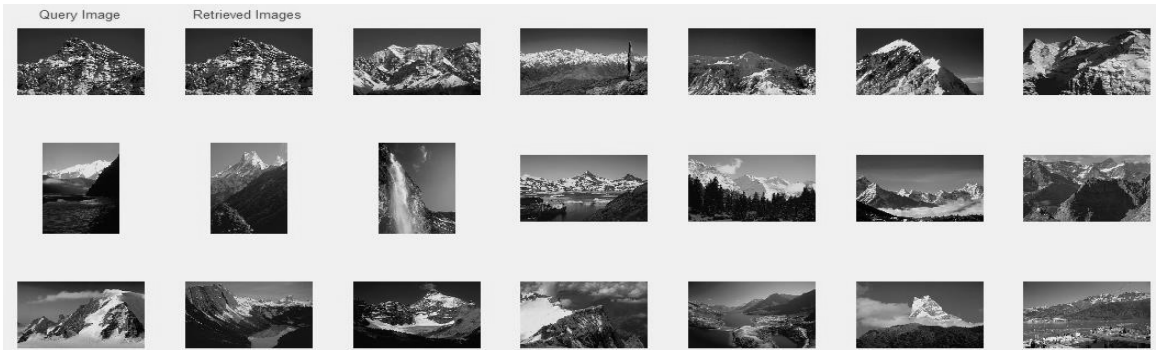

Fig. 3.9 (b)

Fig. 3.9 (a-b): Retrieval result using proposed classification framework on different sample queries. (a). Beach (b). Mountain

Fig.3.9 shows the snapshot of image retrieval for proposed classification framework using RF. Here, image retrieval for same two different Beach and Mountain class images are shown in Fig. 3.9 (a) and 3.9 (b). Using this model, first we extract same discussed features for given query images viz. Beach and Mountain, and fed to trained supervised RF model, and it detects the class of query by aggregating the predictions of various modelled decision tress. After finding the class, searching is automatically narrowed down to detected class images by filtering out all irrelevant classes. As a result, it reliably retrieved all same class images. This approach searching is restricted only in the detected class images, hence average searching time of this approach is 10 times faster than general CBIR system. Because here we narrowed down our search in only 100 images out of 1000 (Wang database have 10 classes having 100 images in each class).

In order to evaluate the performance of this work, we randomly select each images of the database as a query, using K- fold cross validation. This work has taken K=10 for the cross evaluation. It means our extracted dataset is divided into 10 equal sets, each having 100 distinct image instances in which 9 sets (900 image instances-90 images from each class) are used for training and one set (100 image instances-10 images from each class) is used for testing. These processes are repeated 10 times and form a confusion matrix. Using this confusion matrix, we have evaluated various performance measures, which are discussed in tables 3.3 to 3. 6. For the modelling of random forest, the process starts with number choice of decision trees. Fig. 3.10 shows the performance of random forest for different number of trees, where it reflects maximum 83.2 % accuracy for 100, 140, 150 and 200 number of trees. Random forest of 100 trees, each constructed while considering 6 random features takes 0.68 second for building

model for classification, while 140, 150 and 200 number of tree take 0.95, 1.02 and 1.35 second, respectively. So, due to time constraint we have taken 100 numbers of trees for our experimental purpose. From Table 3.3, it is confirm that the performances of this classifier are outstanding for the classes of Bus, Dinosaur, Flowers, and Horses. But Beach class has satisfactorily minimum performances due to strong varieties and more similarity between Mountain class images. Here, 19 % images of this class are misclassified as Mountain and 15 % images of the mountain are wrongly classified as Beach. The overall accuracy of this classifier is 83.2 %, means in 1000 Wang database images; it correctly recognizes 864 images for retrieval.

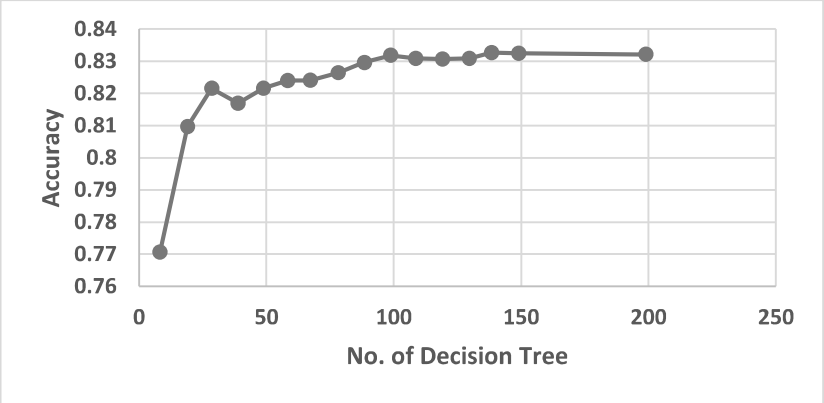

Fig. 3.10: Random forest performance for different number of decision trees

Table 3.3: Confusion matrix for proposed random forests framework (Wang database)

-	Africa	Beach	Building	Bus	Dinosaurs	Elephant	Flowers	Horses	Mountain	Food
Africa	66	2	5	0	1	9	5	2	1	9
Beach	3	66	3	2	0	6	0	1	19	0
Building	3	4	72	8	0	5	1	0	4	3
Bus	0	0	3	95	0	0	0	1	0	1
Dinosaurs	0	1	0	0	99	0	0	0	0	0
Elephant	2	2	2	0	0	86	0	4	3	1
Flowers	4	0	0	0	0	0	96	0	0	1
Horses	0	1	0	1	0	4	1	93	0	0
Mountain	4	15	2	0	0	3	0	0	76	0
Food	4	1	1	4	0	5	2	0	0	83

From the confusion matrix as reported in Table 3.3, it is justified that the Beach and Mountain classes have most correlated images with similar visible appearance, and challenging features are needed to discriminate both classes correctly. From table 3.4, it is clear that the performance of RF is significantly encouraging, we got average precision 83.1 %, recall 83.2 %, MCC=81.2, F-measure= 83.0 and least 1.9 % false positive rate. The overall accuracy of this classifier is 83.1 %, means in 1000 Wang database images; it correctly recognizes 831 images for retrieval. So from all these analysis of performance parameters, it has conveyed that this work performance is better in all areas, shows for classification and retrieval.

Class	Precision	Recall	FP rate	F-Measure	MCC
Africa	0.767	0.660	0.022	0.710	0.682
Beach	0.717	0.660	0.029	0.688	0.655
Building	0.818	0.720	0.018	0.766	0.744
Bus	0.864	0.950	0.017	0.905	0.895
Dinosaurs	0.990	0.990	0.001	0.990	0.989
Elephant	0.729	0.860	0.036	0.789	0.767
Flowers	0.914	0.960	0.010	0.937	0.930
Horses	0.921	0.930	0.009	0.925	0.917
Mountain	0.738	0.760	0.030	0.749	0.720
Food	0.856	0.830	0.016	0.843	0.826
Average	**0.831**	**0.832**	**0.019**	**0.830**	**0.812**

Table 3.4: Classification and retrieval performance for Wang database

- **OT Scene Database Analysis**

For the classification & retrieval analysis of OT database, here, we have taken RF classifier with 70 decision tress, fast in building model. The performances of this work are listed in Table 3.5 and Table 3.6. The performances of this classifier are good for the classes of Forest, Highway and Mountain. But Open Country class has satisfactorily minimum performances due to strong varieties and more similarity between Forest class images. Here, 40 images of this class are misclassified as Forest and 32 images of the Forest are wrongly classified as Open Country, also 39 and 40 images of inside city are wrongly classified as street and tall Building and 21 and 22 images of Tall building are wrongly classified as Inside city and street. These results relate the content similarity between these images.

From Table 3.6, it is clear that the performance of RF is significantly encouraging, we got average precision 75.7 %, recall 75.7 %, MCC=72.1 , F-measure= 75.7 and least 3.6 % false positive rate. The overall accuracy of this classifier is 75.7 %, means in 2688 OT Scene database images; it correctly recognizes 2034 images for retrieval. Also, average searching time of the proposed approach for OT database is 8.22 times faster than general CBIR system

Table 3.5: Confusion matrix for proposed classification framework using random forests classifier on OT database

-	Coast & Beach	Forest	Highway	Inside City	Mountain	Open Country	Street	Tall Building
Coast & Beach	271	1	12	7	14	34	6	15
Forest	2	278	0	0	14	32	0	2
Highway	18	0	206	7	2	15	5	7
Inside City	4	1	8	212	0	4	40	39

Mountain	16	9	3	0	305	35	3	3
Open Country	24	40	6	2	35	286	4	13
Street	4	0	5	39	1	9	219	15
Tall Building	19	1	12	21	9	15	22	257

Class	Precision	Recall	FP rate	F-Measure	MCC
Coast & Beach	0.757	0.753	0.037	0.755	0.717
Forest	0.842	0.848	0.022	0.845	0.823
Highway	0.817	0.792	0.019	0.805	0.784
Inside City	0.736	0.688	0.032	0.711	0.676
Mountain	0.803	0.816	0.032	0.809	0.778
Open Country	0.665	0.698	0.063	0.681	0.622
Street	0.732	0.750	0.033	0.741	0.709
Tall Building	0.732	0.722	0.040	0.727	0.686
Average	**0.757**	**0.757**	**0.036**	**0.757**	**0.721**

Table 3.6: Classification and retrieval performance for OT database

So, the proposed framework outperforms for RF classifier. Furthermore, the feature extraction and searching time of this work is very fast .To extract the feature vectors for the 1000 colour images of size 384*256 (Wang-1000 image database) requires approximately 3 minutes, and for 2688 images of OT scene database of size 256*256, it takes 7.4 minutes. Also for finding the most similar images (in Wang database) to a query image takes 0.31 ms (if the query image belongs to database images) which is extremely promising. Beware the fact that this work searching time is not dependent on image database, it dependent on the class size. So, adding new classes of images will not affect our image retrieval time.

3.3 CBIR System Based on Supervised Learning with Combination of Orthogonal-LBP and Statistical Moments

In this work, a fast and effective CBIR system is proposed which uses supervised learning based image management and retrieval techniques. It utilizes machine learning approaches as prior step for speeding up image retrieval in the large database with enhanced accuracy. For the implementation of this, first we extract statistical moments and the orthogonal combination of local binary pattern based computationally light weighted colour and texture features. Further, using some ground truth annotation of images, we have trained the multi-class SVM classifier. This classifier works as a manager and categorizes the remaining images into different libraries. However, at the query time same features are extracted and fed to the SVM classifier. SVM detects the class of query and searching is narrowed down to the corresponding library. This supervised query image classification and retrieval model filters out maximum irrelevant images and speeds up the searching time. This work is evaluated and compared with the conventional model of the CBIR on two benchmark databases, and it is found that the proposed work is significantly encouraging in terms of retrieval accuracy and response time for the same set of used features.

3.3.1 Methods and Models

In this section, we first describe the feature extraction used in our model. This is followed by the description of our model and how it differs from the conventional one.

3.3.1.1 Feature Extraction

To be able to search similar images, we need some meaningful representation for each image. This is done by mapping the image pixels to a reduced set of the feature vector. As colour, texture, shape and salient points are the key properties by which human eye identifies an image, using these features for the CBIR system is the obvious choice. Our proposed method uses the fusion of two types of the above-listed features which are statistical colour moments extracted from the HSV colour space and a modified version of LBP. These are described as follows

3.3.1.1.1 Statistical Colour Moments

Colour moments are a compact representation of colour distribution that characterise an image. This work has used three colour moments (mean, standard deviation and skewness) as reported in Table 3.7. Further, for better colour indexing of an image, also incorporated two additional statistical moments. These are Kurtosis and Entropy which are dominantly capture the uniformity and other distribution of colour. For each image, we extract a total of 15 colour features, five for each of the HSV component.

Table 3.7: Statistical colour moments

Features	Equation	Descriptions
Mean (μ)	$\mu = \frac{\sum_{i,j} I_{i,j}}{X}$	Reflects average intensity (brightness) of an image I, having X pixels
Standard Deviation (σ)	$\sigma = \sqrt{\frac{\sum_{i,j} (I_{i,j} - \mu)^2}{X}}$	Second order moment shows the contrast of an image.

Skewness (*Sk*)	$\frac{\sum_{i,j}(I_{i,j}-\mu)^3}{X\sigma^3}$	Skewness shows the symmetries of the histogram (distribution) around the mean
Kurtosis (*K*)	$\frac{\sum_{i,j}(I_{i,j}-\mu)^4}{(X-1)\sigma^4}$	Kurtosis shows the flatness of colour intensity
Entropy (*E*)	$-\sum_{i,j} I_{i,j}\log_2 I_{i,j}$	The Entropy represents the uniformity of colour intensity

3.3.1.1.2 Orthogonal Combination of Local Binary Patterns (OCLBP)

In this work, we have incorporated a modified version of LBP called the orthogonal combination of local binary patterns (OCLBP). For LBP histogram where eight neighbours are mostly used, produces a rather long (256-dimensional) histogram. The size of the LBP histogram will significantly reduce to 16 if only four neighbouring pixels are taken into account. As compare to LBP, OCLBP reduces the feature dimension from 2^P to $2\times2^{P/2}$, where P is the neighbouring pixels.

Working of OCLBP is shown in Fig. 3.11, where $OCLBP1(i,j)$ represents the LBP value calculated for the pixel at location (i,j) in the image matrix using only its neighbouring vertical and horizontal pixels. Whereas, $OCLPB2(i,j)$ represents the same for the neighbouring diagonal pixels. The final $OCLBP$ is obtained by concatenating histograms of $OLBP1$ and $OLBP2$. OCLBP not only reduces the dimension of LBP histogram from 256 to 32 but also preserves the discriminative power of LBP as the number of distinct binary patterns is same ($2^4 \times 2^4 = 2^8$). In Fig. 3.12, the sample image of mountain class from COREL database is taken, and OC-LBP histograms are computed from diagonal pixels and horizontal-vertical pixels. After concatenation of these histograms, we form OC-LBP features. Same way Tall-Building class from Oliva and Torralba database are taken and histograms are computed using

diagonal pixels, and horizontal-vertical pixels. Mathematical model and extraction steps for the OC-LBP feature is given in the next section.

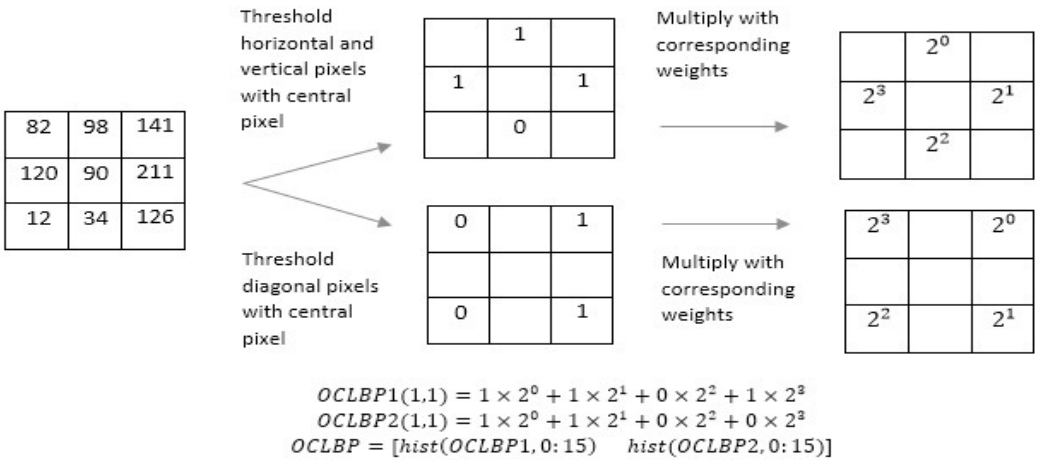

Fig. 3.11: OCLBP Feature

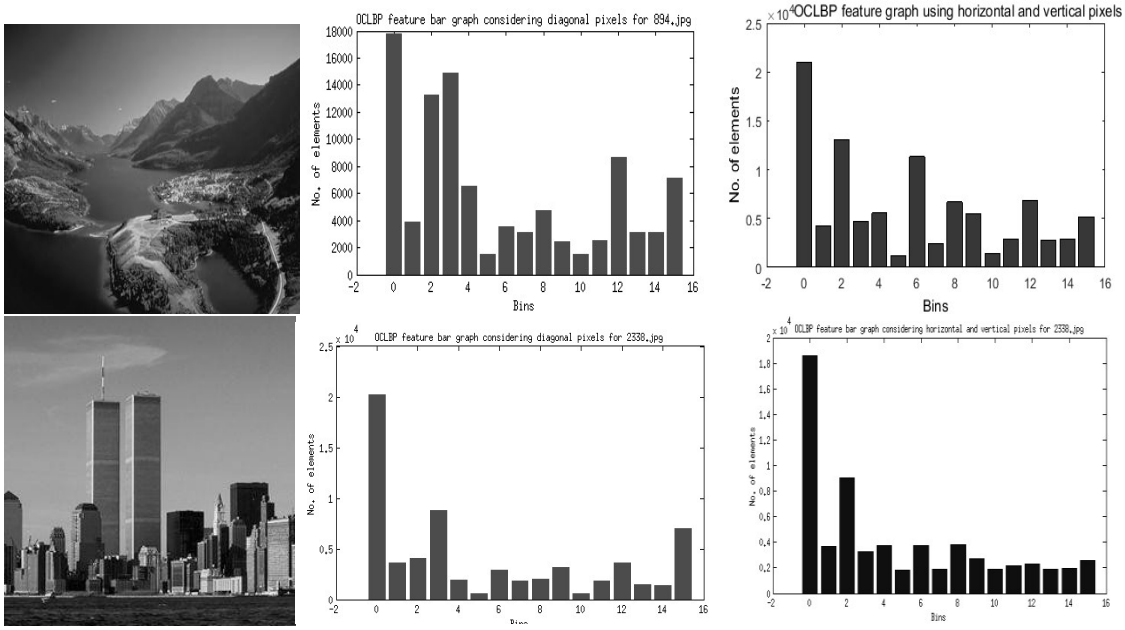

Fig. 3.12: OCLBP Histogram for different sample images

3.3.1.2 Proposed Models

The proposed framework is based on the detection of query images because correct classification and detection of query images greatly enhance the performance of retrieval by filtering out irrelevant images during matching. For query image

classification and retrieval model, it is highly desirable to make correct groups of meaningful categories based on their low- level features. Here in this work, to achieve the goal of automatic categorization of the images, we have proposed supervised learning based model using SVM, which classifies the images based on the pre-learning from some ground truth images. Proposed CBIR system is divided into two parts whose working diagram is shown in Fig. 3.13 and Fig. 3.15. In Fig. 3.13, images are grouped into different clusters based on trained SVM classifier. At the beginning, system provides *k%* annotations for each class images as a ground truth. SVM classifier will be trained based on this ground truth information. A classifier will predict annotation for remaining *(100-k)* % images that are not part of the ground truth based on current ground truth. So, the remaining images are grouped into different clusters according to the SVM prediction. For the training of SVM model, a fusion of colour moments and the orthogonal combination of LBP features are used whose extraction procedures are given in Algorithm 3.2.

Algorithm 3.2: Feature Extraction

Input: Images loaded from database

Ensure: CM() and OCLBP() are proposed functions to extract statistical colour moments and orthogonal-LBP features from the image, which extraction functions are given in Algorithm 3.2.1 and Algorithm 3.2.2.

Output: Features stored in file f

Steps

1. begin
2. K←total no. of images
3. Set feature matrix x=[]
4. for i=1 to K

5. x=[x; CM(images{i}), OCLBP(images{i})]
6. end for
7. save it in file f
8. end

Algorithm 3.2.1: Statistical Colour Moments (CM) Extraction

Input: Image I

Output: Colour feature vector CM of length 15

Steps

1. begin
2. Set R ← Red colour channel of I
3. Set G ← Green colour channel of I
4. Set B ← Blue colour channel of I
5. for each R,G,B
6. calculate μ, σ, S_k, K, and E statistical features
7. append calculated features to CM
8. end for
9. end

Algorithm 3.2.2: OCLBP Texture Feature Extraction

Input: Image I.

Ensure: grayscale() converts RGB image to grayscale. Function P(x) outputs 1 for positive x, and is otherwise 0. histogram(h) arranges h in bins of size 16 and returns a vector of length 16 containing number of elements in each bin.

Output: Texture feature vector OCLBP of length 32

Steps

1. begin

2. Set M=rows(I), N=columns(I)
3. Set g=grayscale(I)
4. for i=2 to M-1
5. for j=2 to N-1
6. J0←g(i,j)
7. J1←vertical and horizontal neighbouring pixels
8. J2←diagonal neighbouring pixels
9. $g1(i,j) = \sum_{k=0}^{length(J1)} P(J1(k) - J0) \times 2^k$
10. $g2(i,j) = \sum_{k=0}^{length(J2)} P(J2(k) - J0) \times 2^k$
11. end for
12. end for
13. OCLBP ← [histogram(g1), histogram(g2)]
14. end

Finally, for indexing these entire feature vectors are stored in the database along with the images and clustering of images are formed using this feature set. Suppose, we have a dataset of K number images such that each image belongs to one of the n classes ; $C_1, C_2, C_3, \ldots C_n$. For each image, a total of 47 features is extracted and stored in a feature vector. To ensure equal weightage is given to all features, these features are normalized after extraction-each feature is divided by its range.

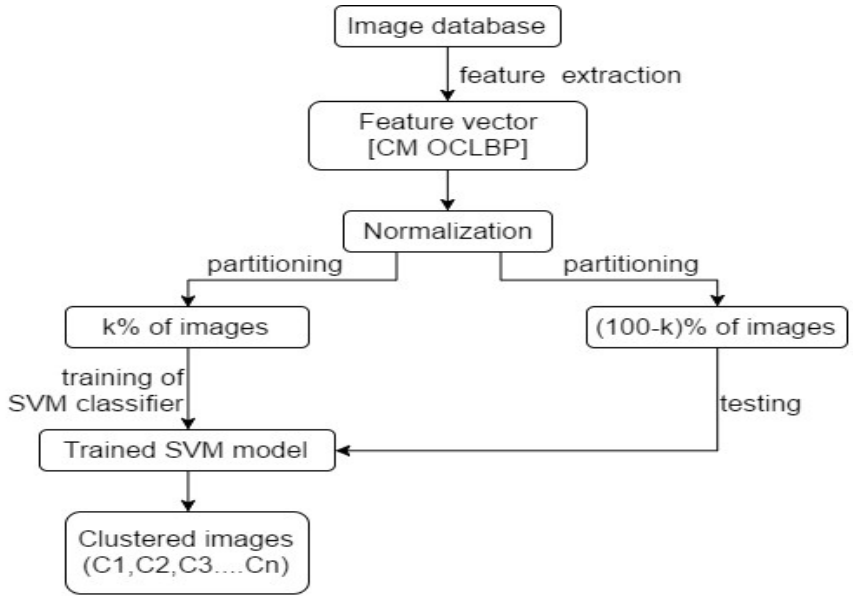

Fig. 3.13: Proposed supervised learning based pre-clustering framework

As we know, $k\%$ of images from each class is used as ground truth. Next, a SVM classifier is trained with the feature vectors of these images as input. A suitable kernel is chosen and its parameters are tuned. The trained classifier then predicts the class for each unlabelled image. Finally, for each class a separate cluster (folder) containing images predicted for that class is created. This sets up our image database. For instance, in our datasets K was intermediate and size of feature vector is small. Hence, a RBF kernel was selected. We used LibSVM to implement multi-class image classification [137]. Only k=20% of the K images were used for training. The parameter gamma of radial basis function was tuned with the help of 4-fold cross validation. Gamma was plotted against accuracy as shown in Fig. 3.14 to find the optimal gamma. Using optimal gamma SVM model is trained. Finally, Prediction was done on rest of the unknown 80% images in the dataset and assigned an appropriate cluster $(i.e.\ C_1, C_2, C_3, \dots C_n)$ to each database images. Fig. 3.15 shows proposed retrieval model, where for given query image, we extract its features (as discussed above) and normalize each feature using its range. The trained SVM classifier then

predicts the class of this image. Next, images are retrieved from the cluster of the predicted class based on similarity criteria which assigns a rank to each image. The top images obtained from this procedure are then returned. In our case, ED between the extracted features of the query image and that of the images of the predicted folder was used as similarity criterion. The complete pre-processing and retrieval model of this work is also presented in algorithm 3.3.

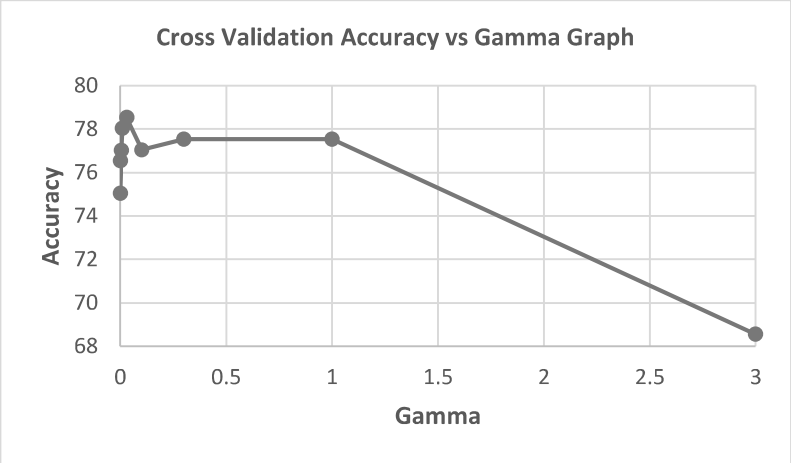

Fig. 3.14: Tuning of gamma for training the classifier

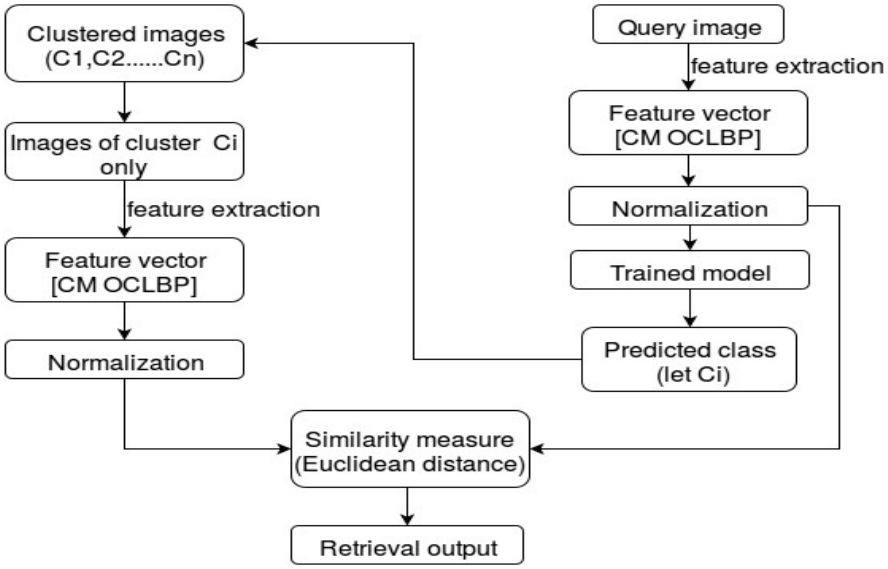

Fig. 3.15: Proposed image retrieval framework using supervised learning

Algorithm 3.3: Image Retrieval

Input: Features file f.

Ensure: f contains features x from image database. loadFile loads feature variables from file f. scaleFeatures() normalizes the features x. svmTrain() trains a model on training data based on the training parameters. svmPredict() predicts labels for test data using trained model. euclideanDistance() calculates the distance between input feature vectors.

Output: Top images T retrieved for each of the input image features chosen randomly.

Steps

1. begin
2. x←loadFile(f)
3. Initialize n=columns(x), numClass ← total no. of classes of images
4. X ← scaleFeatures(x);
5. for i = 1 to numClass
6. Xtrain ← feature vector of 20% of images from class i
7. ytrain ← image labels corresponding to Xtrain
8. Xtest ← feature vector of remaining images from class i
9. ytest ← image labels corresponding to Xtest
10. end for
11. parameters ← find optimal value of parameters C and gamma
12. model = svmTrain(ytrain, Xtrain, parameters);
13. [ypredict] = svmPredict(ytest,Xtest,model);
14. send the images to the folder of their predicted class
15. for i = 1 to numClass
16. Xinput ← feature vector of random 10% images which are in Xtest

17. yinput ← image labels corresponding to Xinput

18. end for

19. pred = svmPredict(yinput,Xinput,model);

20. for i = 1 to length(yinput)

21. folder = pred(i)

22. d ← images in folder

23. for j = 1 to d

24. ed(j) = euclideanDistance(image{i},image{d(j)})

25. edSorted = sort(ed)

26. T(i) ← top images from edSorted

27. end for

28. end for

29. end

In the conventional model, given a query image similarity between the normalized features vectors of the query image and each of the database images is simply calculated based on some criterion. Images are retrieved from the entire dataset based on similarity. It can be observed that our model is similar to the conventional one. However, a major step at which it differs is that of the query image classification and image retrieval from a predefined cluster only. In the traditional model, once normalized feature vectors of images are obtained, direct matching of similar images from the database is done. In contrast, in our model we are first predicting the image class and then searching similar images from the set containing images predicted in the same class. This set of images is not only smaller in size as compared to the whole image database but also contains the higher proportion of similar visual content. Notice that, in our model, it is required for only a small proportion of the image set to be

labelled. Also, in actual CBIR systems, the labelled images should also be sent to respective class sets. Although this would result in higher retrieval accuracy, we restrict ourselves from doing this to compare our approach to the conventional one in an unbiased manner.

3.3.2 Result Analysis and Discussions

We evaluated our model on two benchmark databases viz. Wang and OT database. In our experiments, we used 20% ground truth images of each category for the training of the SVM classifier. The trained classifier then predicted the class of the remaining 80% of images with some classification accuracy. Let this classification accuracy be clustering classification accuracy (CCA). For querying our CBIR system, we randomly took 10% sample images of each category from the remaining 80% of image database as input. Let the classification accuracy of these images be query classification accuracy (QCA). For each input image, 50 images were retrieved from the predicted library.

To compare our method with the conventional approach, we implemented the conventional model (as Fig.1.1) on the same datasets taking the same sample query images used to test our method. Identical normalized feature vectors were extracted from all the images in the dataset. These vectors were matched with the normalized feature vector of query image based on Euclidean distance and topmost similar images were retrieved. To test whether performance is feature dependent, we also took both colour moments and OC-LBP alone as features and repeated the whole process of retrieval for the proposed and conventional approaches.

3.3.2.1 Performance on Wang (COREL) Database

CCA for this dataset was 82.25%, causing 658 images out of a total of 800 images to be sent to correct libraries. The corresponding confusion matrix is shown in table 3.8. From the confusion matrix, we can see the classification performance of all classes where CCA of Bus and Dinosaur classes are outstanding, only one and two images of these classes are misclassified into different libraries. QCA was 88%, causing 88 query images out of 100 to be predicted correctly which is very promising. Fig. 3.16 (a-d) show the retrieval performance on different sample queries, where for conventional framework query image are searched in the entire database. But for the proposed framework, adopted SVM performs the classification with different class-wise CCA, in which searching is performed. For given query image (African and Elephant), the low level features are extracted and it is given as input to the already trained SVM model.

Category	African	Beach	Monuments	Bus	Dinosaur	Elephant	Roses	Horse	Mountain	Food
African	64	1	0	0	1	6	4	3	0	1
Beach	2	60	0	0	0	5	1	3	9	0
Monuments	6	4	49	1	0	6	1	5	5	3
Bus	0	2	4	74	0	0	0	0	0	0
Dinosaur	0	1	0	0	79	0	0	0	0	0
Elephant	7	10	0	0	0	56	0	2	5	0
Roses	3	0	0	0	0	0	76	0	0	1
Horse	1	1	0	0	0	1	0	75	0	2
Mountain	0	6	0	0	0	6	2	8	56	2
Food	1	2	0	1	0	1	6	0	0	69

Table 3.8: Confusion matrix of proposed supervised model for Wang database

Fig. 3.16 (a) Conventional retrieval for African class

Fig. 3.16 (b) Proposed retrieval for same African query

Fig. 3.16 (c) Conventional retrieval for elephant class

Fig.3.16 (d) Proposed retrieval for same elephant class

Fig. 3.16 (a-d): Sample retrieval for African and Elephant using conventional and proposed method

The trained SVM predicts the category of the query image which is nothing but the semantic concept of the query image. Hence instead of finding similarity between the query image and all the images in the database, it is found between the query image and only the images belonging to the query image category. So, the retrieval results are more accurate. Fig. 3.17 (a-b) show the precision and recall from both approaches using different features when 50 images for each input query is retrieved. The blue bar is for the proposed method and red one is for the conventional one. In these figures, abbreviation CM denotes the statistical colour moments, OCLBP is orthogonal combination of LBP and last one is the fusion of both. The graphs depict that our method is much better than the conventional one for all classes in this dataset. This improvement is class and feature independent. However, in rare cases, such as for bus (class 4) when only statistical colour moments (CM) are used, the precision values are low for the proposed method as compared to the conventional one. This is due to the classifier predicting incorrect class for most of the query images of a particular class. A match is said to be occurred only if the class of both the query and retrieved image is same. From the graph Fig.3.18 (a-b), it is observed that proposed framework is significantly encouraging than the conventional framework for the same independent feature sets viz. statistical colour moments, OCLBP and combination of both. In this analysis, number of retrieved images from 1 to 50 are taken and calculated the respective average precision and recall values. In this graph P and C denotes the proposed and conventional framework. Again it can be inferred that improvement in performance is feature independent. For any set of independent feature, the proposed work is more promising than the conventional approach. As the number of images increase, the performance of the conventional method deprecates much more drastically

as compared to the proposed method with its precision value becoming almost constant.

So, proposed method performs way better for a large number of images.

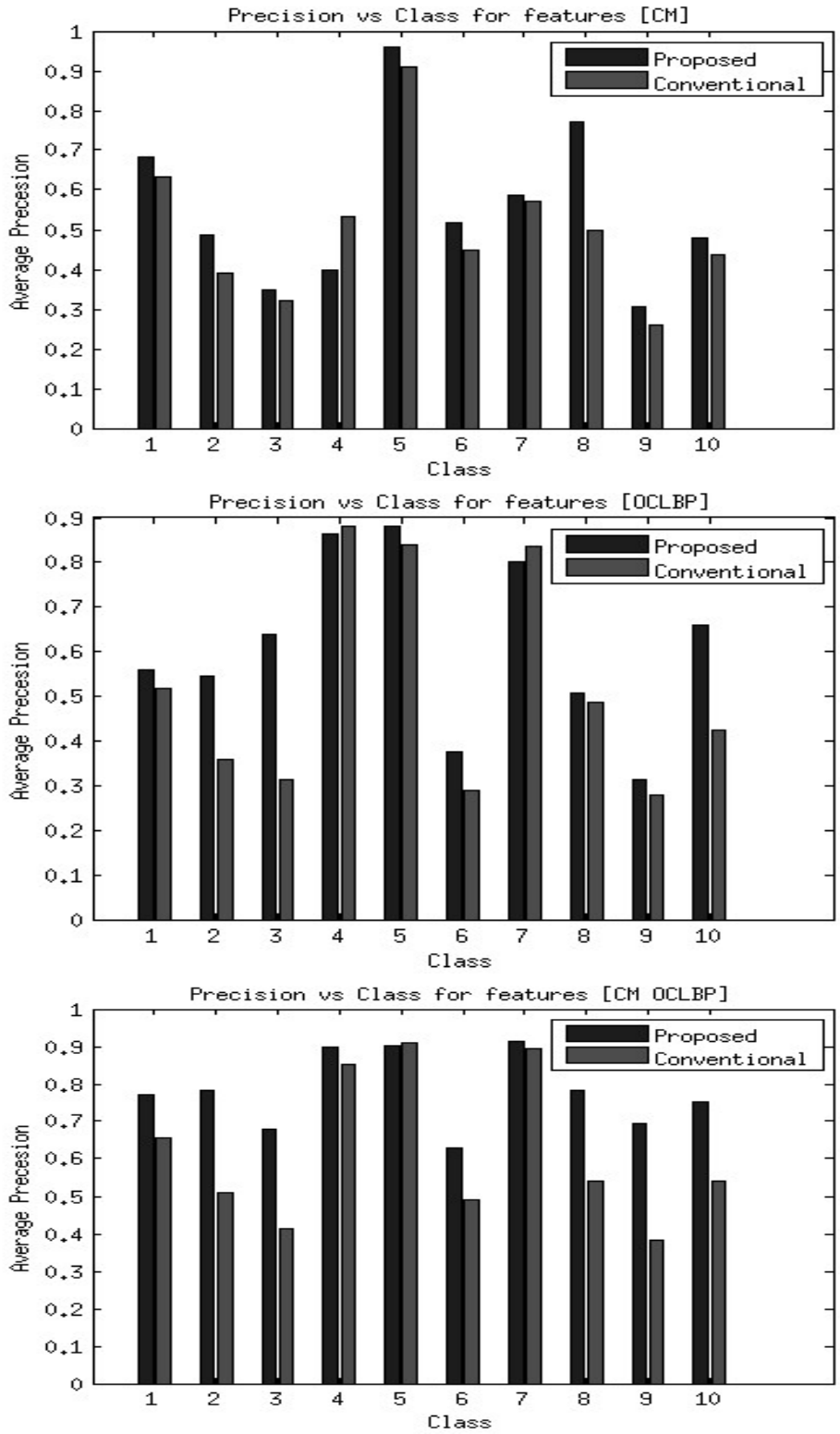

Fig.3.17: (a) Precision vs Class bar graph for retrieval of 50 images per query with different feature vectors.

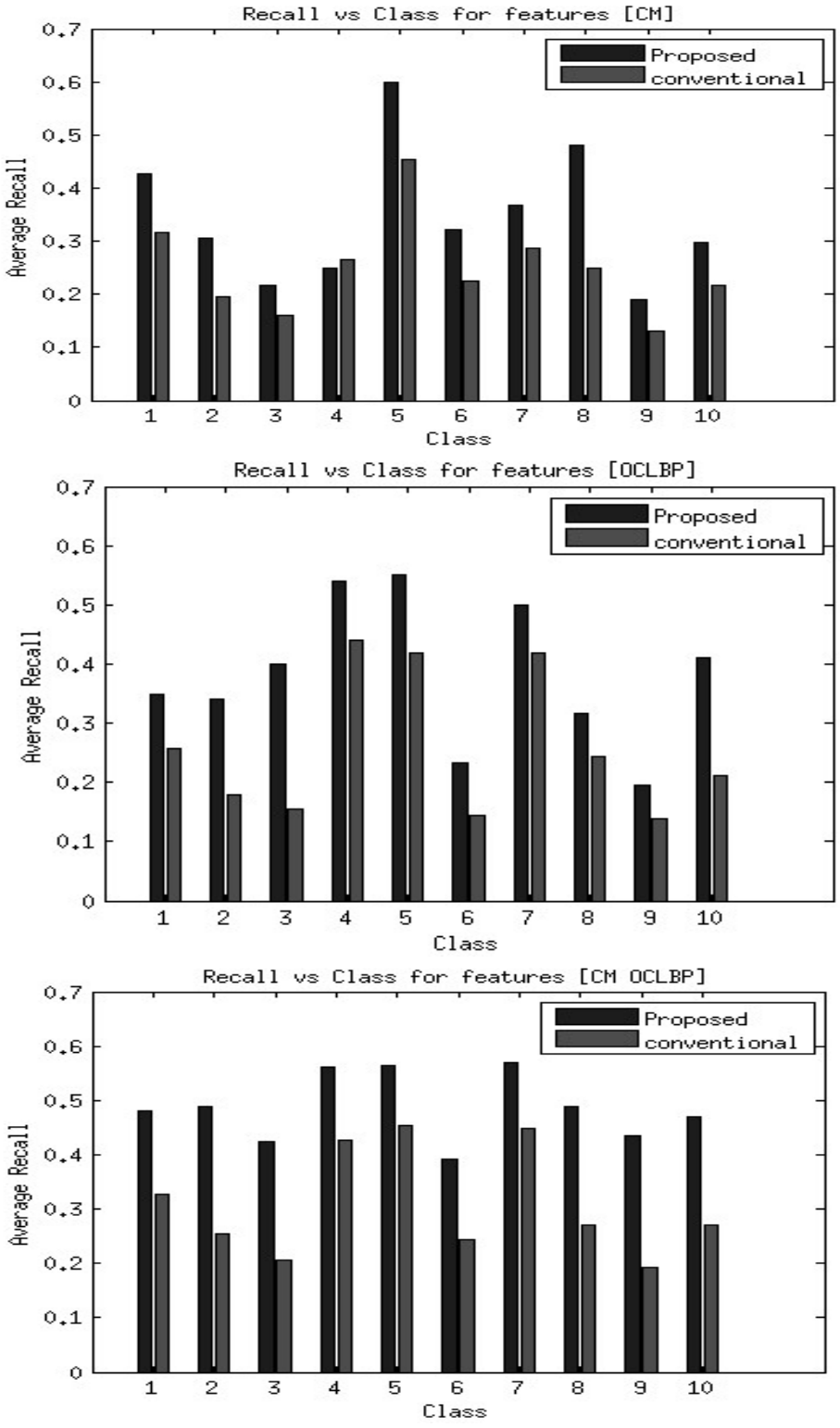

Fig.3.17: (b) Recall vs Class bar graph for retrieval of 50 images per query with different feature vectors

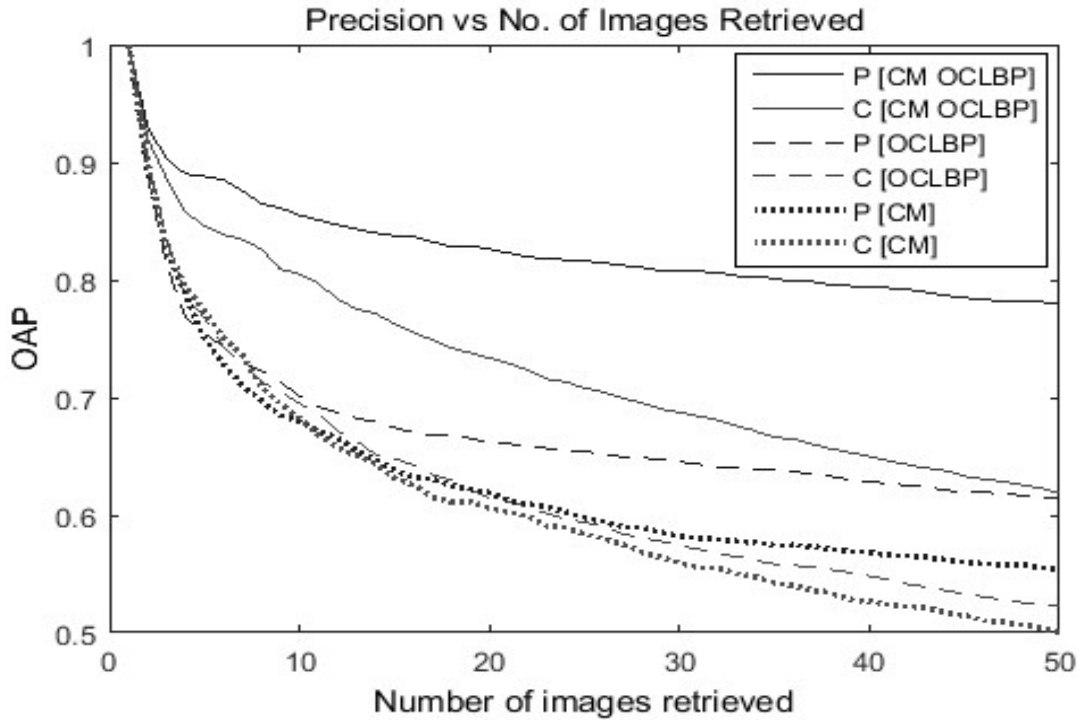

Fig. 3.18(a)

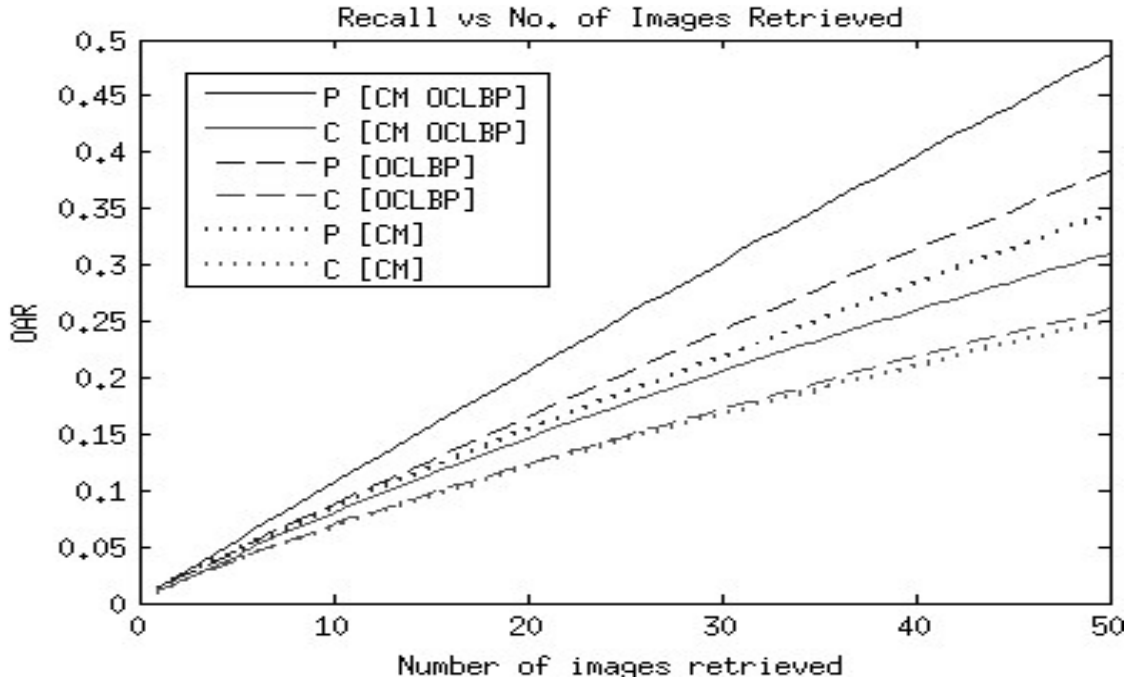

Fig. 3.18(b)

Fig. 3.18 (a): Precision vs. No of images retrieved, (b) Recall vs. No of images retrieved

Table 3.9: Comparing with different conventional approaches for Corel Database

Methods	Classes										
	Africa	Beach	Building	Bus	Dinosaur	Elephant	Flower	Horse	Mountain	Food	OAP
Walia and Pal [9]	0.50	0..88	0.56	0.73	1.00	0.84	0.99	1.00	0.77	0.37	0.764
PS and Pujari [59]	0.48	0.34	0.36	0.61	0.95	0.48	0.61	0.74	0.42	0.50	0.549
Yue *et al.* [60]	0.59	0.41	0.43	0.72	0.75	0.65	0.83	0.69	0.45	0.46	0.595
Jalab [61]	0.32	0.61	0.39	0.40	1.00	0.56	0.89	0.65	0.56	0.44	0.582
Shenet *et al.* [62]	0.68	0.65	0.52	0.97	0.99	0.52	0.83	0.78	0.61	0.73	0.728
Rahimi *et al.* [64]	0.42	0.45	0.49	0.31	0.55	0.70	0.61	0.57	0.44	0.44	0.497
Liu *et al.* [65]	0.54	0.51	0.38	0.46	1.00	0.63	0.90	0.93	0.48	0.39	0.518
Zhao et al[67]	0.70	0.66	0.62	0.88	0.84	0.82	0.90	0.78	0.64	0.85	0.769
Mistry et al [68]	0.71	0.72	0.66	0.81	0.91	0.63	0.87	0.68	0.65	0.73	0.737
Proposed	0.845	0.84	0.68	0.91	0.905	0.665	0.975	0.895	0.725	0.82	**0.826**

In Table 3.9, we have compared this work with other latest conventional approaches, where the focus is on feature indexing and similarity measures. In this study, only 20 retrieved images are taken. From the class-wise average precision and overall average precision (OAP), it is clear that the proposed work is significantly encouraging than other methods. Also, the response time of this work is very fast, which is estimated using speedup metric.

Table 3.10: Retrieval speedup for Wang database

Category	Number of Similarity Comparisons	Speedup
African	84	11.90
Beach	87	11.49
Monuments	53	18.87
Bus	76	13.16
Dinosaur	80	12.5
Elephant	81	12.35
Roses	90	11.11
Horse	96	10.42
Mountain	75	13.33
Food	78	12.82
Average	-	12.79

Searching time improvement, as compare to conventional CBIR based exhaustive search methods is estimated using Speedup metric, given in Table 3.10. Here, we got the maximum *Speedup* up to 18.87. Means, searching time is 18.87 times faster than conventional CBIR (if the query image belongs to the monuments cluster having a minimum number of images). And the minimum searching *Speedup* is 11.49 times, if the query belongs to the Beach class, having a maximum number of images in the cluster. The number of images in the corresponding libraries (cluster) depends upon the classifier performance which is the sum of predicted true positive and false positive images, as reported in confusion matrix. The overall searching time improvement as compared to convention CBIR is 12.79. So, this work retrieval performance is very promising with fast and accurate response for a particular query.

3.3.2.2 Performance on Oliva and Torralba (OT) Database

CCA for this dataset was 64.3689%, causing 1382 images out of a total of 2147 images to be sent to correct libraries which is good when high dimensionality of close pictures and number of classes are considered. Also, distribution of this database is non-uniform.

Table 3.11: Confusion matrix of proposed supervised model for OT database

Category	Coast & Beach	Open Country	Forest	Mountain	Highway	Street	City Centre	Tall Building
Coast & Beach	165	5	64	16	13	17	6	1
Open Country	0	205	0	0	19	36	2	0
Forest	30	3	127	3	5	9	26	5
Mountain	0	4	3	176	0	0	47	16
Highway	24	27	2	0	196	47	2	1
Street	23	31	12	5	32	197	25	3
City Centre	2	5	12	23	3	0	168	20
Tall Building	3	23	5	23	14	2	66	148

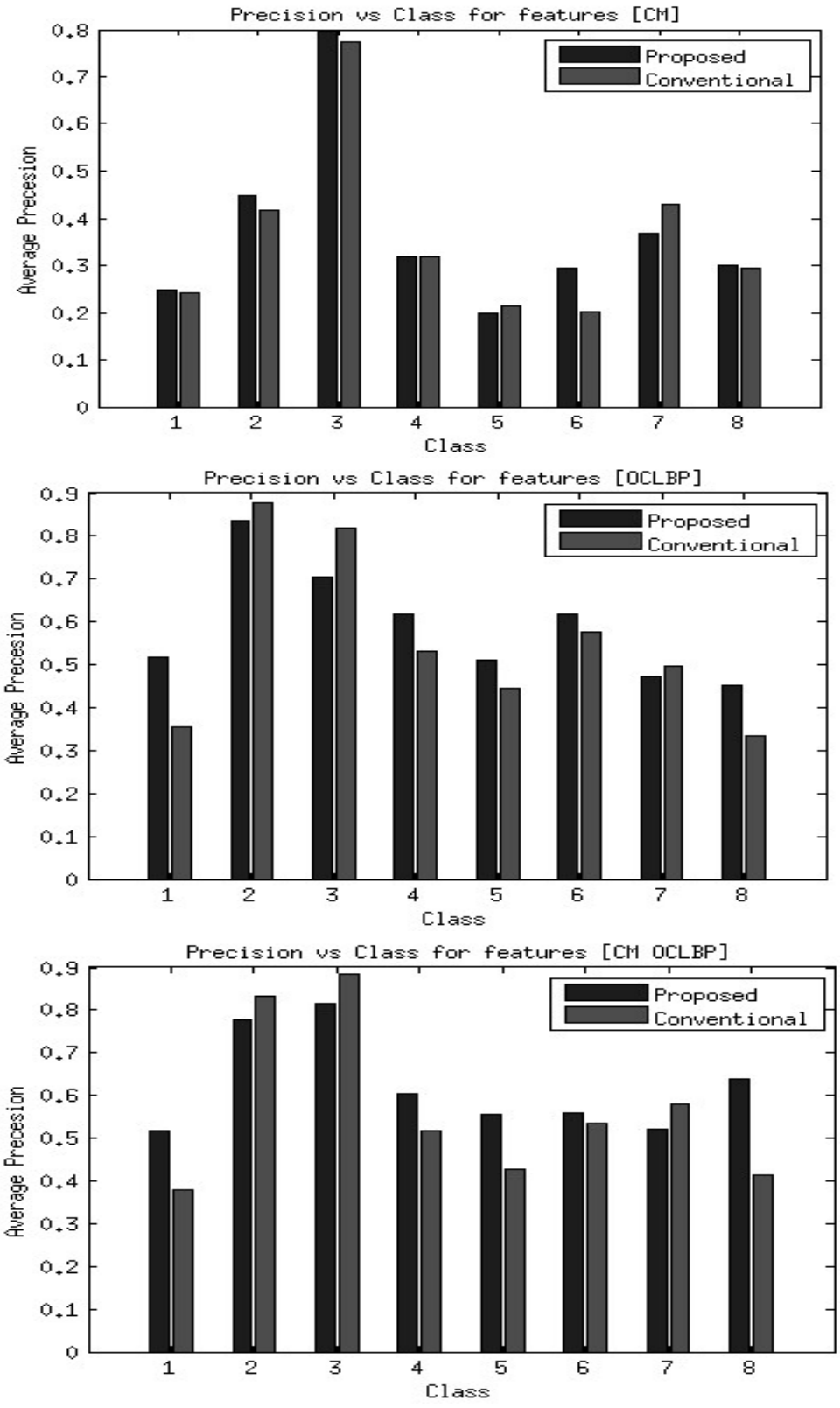

Fig.3.19 (a): Precision vs Class bar graph for retrieval of 50 images per query with different feature vectors

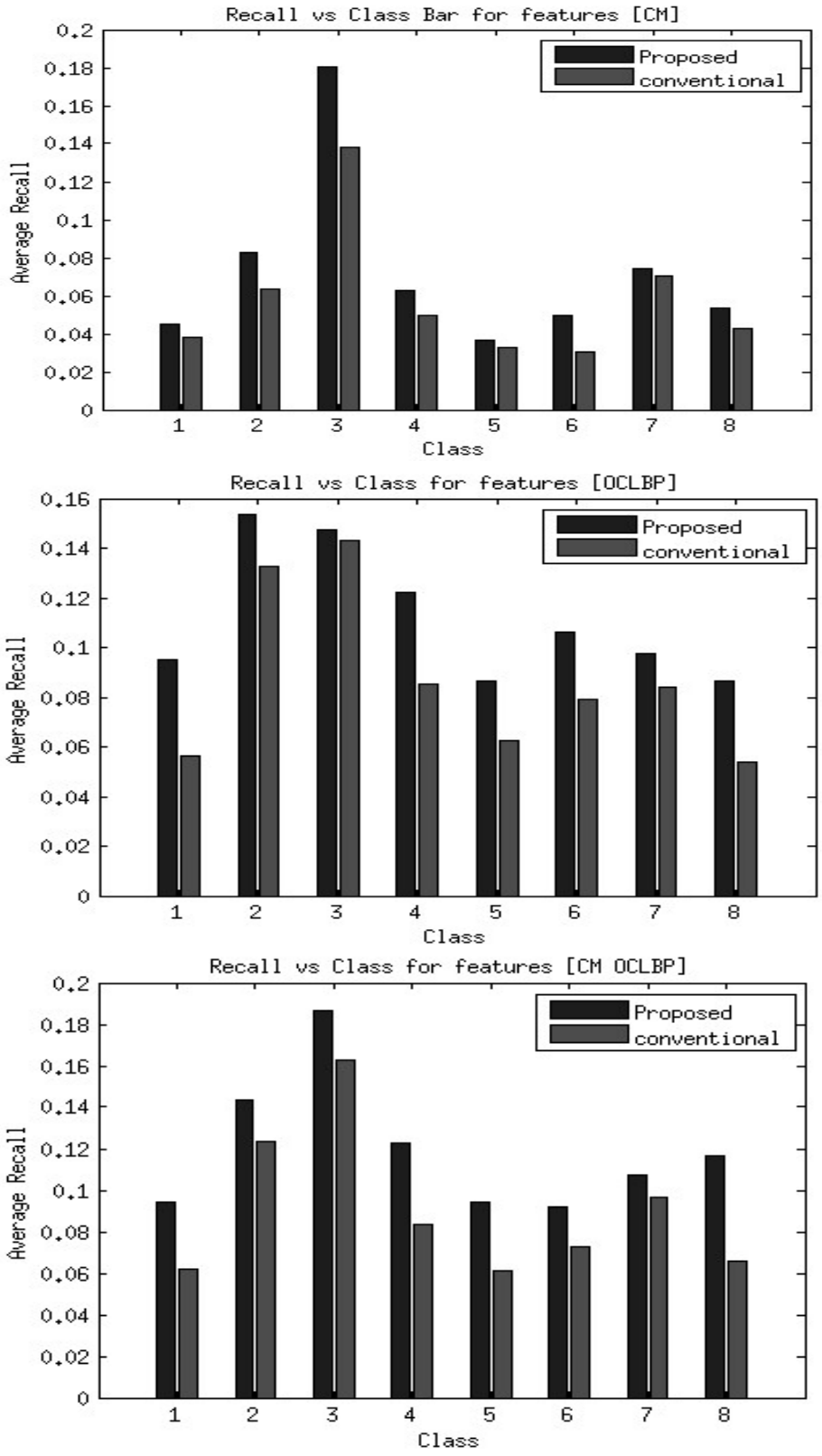

Fig.3.19 (b): Recall vs Class bar graph for retrieval of 50 images per query with different feature vectors

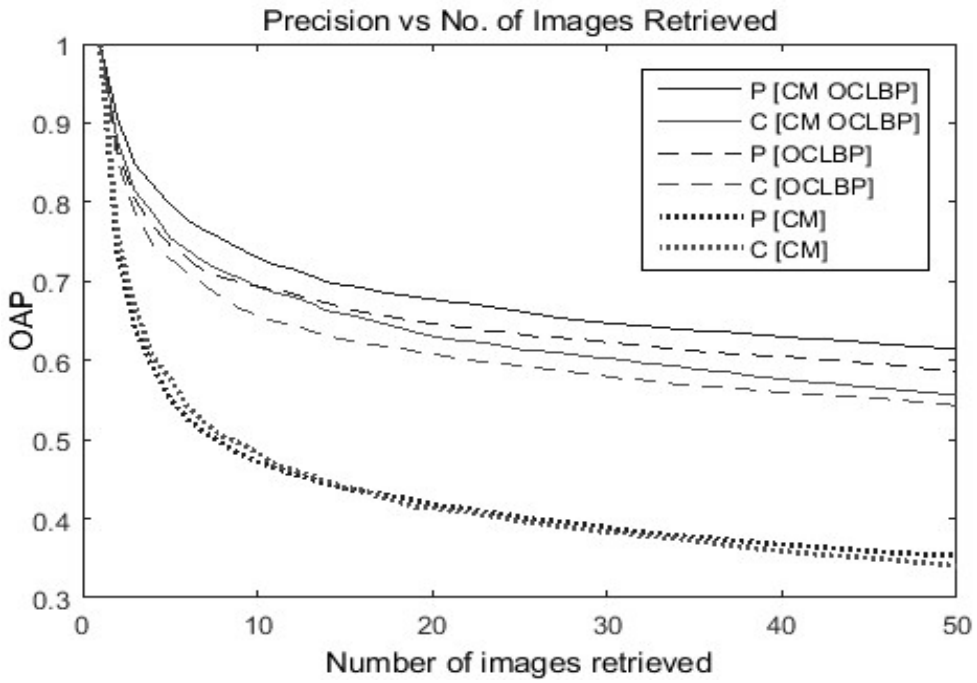

Fig.3.20 (a)

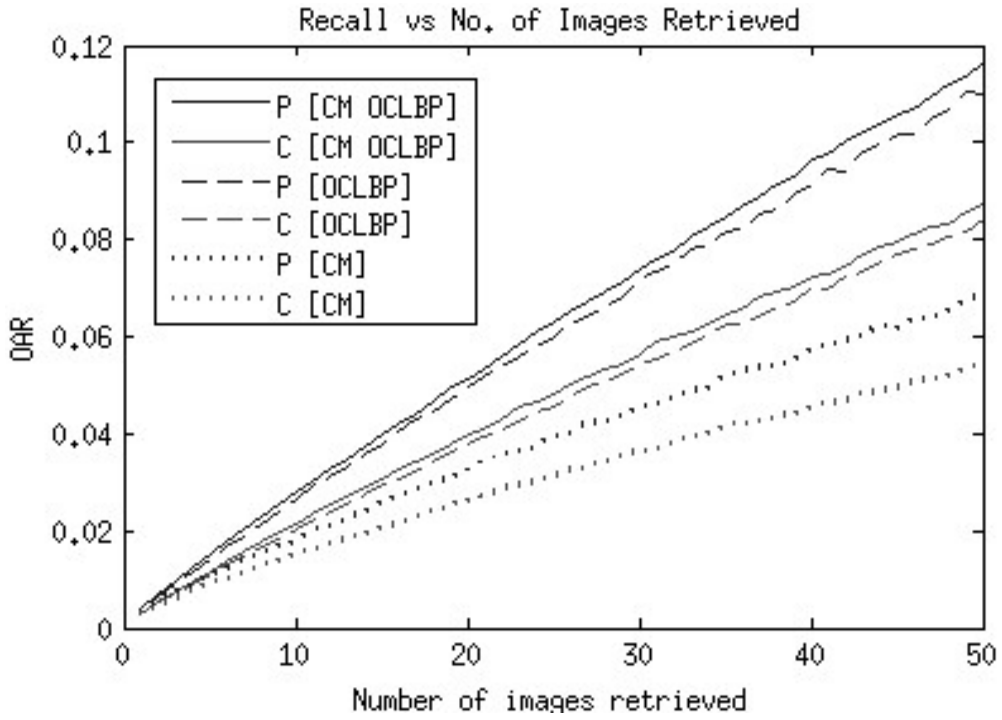

Fig. 3.20(b)

Fig. 3.20: (a) Precision vs. No of images retrieved, (b) Recall vs. No of images retrieved

The corresponding confusion matrix is shown in table 3.11. From the confusion matrix, we can see that there are maximum 66 images of the tall building are misclassified into city centre library. Also, 64 images of the coast-beach class are miss-classified into forest. This is due to the strong visual correlation between both class images. QCA was 69.75%, causing 189 query images out of 271 to be predicted correctly. Fig.3.19 (a-b) shows the variation of precision and recall with the class and features used. Again, for this database, it can be seen that performance enhancement is class and feature independent. However, the improvement is less as compared to the COREL database. This is due to the fact that this database contains many images which are visually similar to some classes it does not belong to. Fig. 3.20 depicts graphs of precision and recall as the number of images varies. Similar to the graphs Corel database, the performance of the proposed method becomes much better for larger number of images. The performance deprecates at a much larger rate for the conventional method.

Table 3.12: Retrieval Speedup for OT Database

Category	Number of Similarity Comparisons	Speedup
Coast and Beach	247	10.88
Open Country	303	8.87
Forest	225	11.95
Mountain	246	10.93
Highway	282	9.53
Street	308	8.73
City Centre	342	7.86
Tall Building	194	13.86
Overall	-	10.33

Table 3.12 reflects the searching time improvement for OT database, here, we got the best *Speedup* up to 13.86 times faster than conventional CBIR, if the query image belongs to the tall building (have a minimum number of images) cluster. And minimum searching *Speedup* is 7.86, belong to the City class, having a maximum number of images in the cluster. The overall searching time improvement as compared to convention CBIR is 10.33 times, which are quite promising for this un-uniform database. Thus, the proposed framework outperforms for the retrieval. Further, the feature extraction and searching time of this work is very fast. To extract the feature vectors for the 1000 colour images of size 384*256 (Corel-1000 image database) requires approximately 129 seconds, and for 2688 images of OT scene database of size 256*256, it takes 247 seconds. Beware the fact that this work searching time is not dependent on size of image database, it depends on the class size. So, adding new classes of images will not affect our image retrieval time. Also, this framework is adaptive and generalized for any set of features. It gives encouraging results as compared to conventional model for the same set of used features. Further, narrow down of search space during image retrieval tend to this system can be easily scalable on big database.

3.4 Conclusions

This chapter demonstrated simple and effective approach for image retrieval based on fusion of fast feature sets, weighted similarity measure and machine learning approaches. This chapter contribution was divided into two sections. In first section, varying weighted similarity measure using conventional approach gives encouraging retrieval performance. Further, this contribution was extended on pre-clustered images

and trained the random forests using all ground truth of database which narrowed down the search space and got promising test results.

In the second section, fast and effective image retrieval was proposed which uses light weighted colour and texture features in the supervised environment. SVM classification based machine learning approach was incorporated which was a pre-pressing step for speeding-up image retrieval in large databases. This trained supervised model using some ground truth improved the retrieval accuracy and searching time by filtering out all irrelevant images of the database.

In both contributions, used feature sets are compact with rich information, fast in indexing and constant in size. In addition, as opposed relationship between low-level features and higher level semantics viz. object and their inter-relationships, this chapter resolved the semantic level problem by applying the supervised machine learning techniques with different variants of colour and texture features. Also doing a linear search on the entire image database takes a lot of time and to increase searching efficiency, both models searching space is narrowed down in a particular reduced set of images which the query is predicted to be a part of. Both contributions narrowed down the search space and got promising retrieval results as compared to the conventional and other state-of-art methods. Therefore, using supervised learning based techniques; we not only improved the efficiency of the CBIR systems but also improved the accuracy of the overall process.