Content Based Image Retrieval through Materialized View

Debabrata Datta[1], Anal Acharya[2], Kashi Nath Dey[3]

[1]Department of Computer Science, St. Xavier's College (Autonomous), Kolkata, India, debabrata.datta@sxccal.edu
[2]Department of Computer Science, St. Xavier's College (Autonomous), Kolkata, India, anal.acharya@sxccal.edu
[3]Department of Computer Science and Engineering, University of Calcutta, Kolkata, India, kndey55@gmail.com

Abstract - A materialized view is a database object which physically stores the output of a query or even of a query set. This database object is primarily used for faster query execution as well as for data analysis. Materialized views can well be used for acquiring a comprehensively fast response to the analytical queries. This is why the materialized views are increasingly used in commercial applications. Most of the applications of materialized view put forward so far have dealt with text data but this paper proposes an algorithm to apply materialized view in the field of image classification as well as retrieval and to do so the proposed algorithm tries to extract some defining features of images to identify their classes. Hence, instead of storing the entire images, this research work introduces a method to store some important parameters related to images in the form of materialized views and takes the help of content based image retrieval approach. In the process, the proposed algorithm utilizes the fast accessing nature of a materialized view and at the same time, the amount of memory space required to identify the class of an image has also been reduced. Effectively, both the space constraint and the time constraint are taken care of for the content based image retrieval operation.

Keywords:
materialized view, content based image retrieval, supervised learning, support vector machine, image classification

1. Introduction

A view or often termed as a logical view is a database object which temporarily stores the result of a query. On the other hand, a materialized view is also a database object but unlike a logical view, it physically stores the output of a query or even of a query set [1]. Many commercial database packages have, of late incorporated the generation of a materialized view and the use of the same [2][16]. Materialized views help in eliminating the overhead involving expensive joins and aggregations that need to be performed for a large class of queries and offer significant improvements in query processing time, particularly for aggregation queries over tables of considerable size [3][4][15]. There are different approaches of identifying the optimal ways of storing data to be materialized. In general, the use of materialized view lies in the field of faster query processing and a methodical data analysis. Since most of the areas where materialized view is often used deal with text-based data, there are different ways or algorithms to do that. This research paper proposes a method of utilizing materialized view in image classification and retrieval which find applications in a wide variety of different domains. Because of the exponential growth in the field of multimedia applications, storing and retrieving the desired digital data in the form of images have become an area of concern, of late. The goal of the proposed method is to improve the way the images may be stored, retrieved based on their different classes. Image classification is the process of classifying images into one of two or more classes based on certain features extracted from the images. Retrieving images based on their defining features leads to the concept of Content Based Image Retrieval or CBIR which is the process of searching and retrieving images from a database or collection of images based on an image query. In this research paper, a method has been proposed to utilize materialized view in the domain of content based image retrieval. Here, the image query consists of the features of the images to be classified and they are compared against the features of the images stored in the form of materialized views. Instead of retrieving matching images as is the case in the traditional CBIR, the new images are added to the materialized view that contains the matching images. So, the main objective of the proposed research work is to classify the unclassified images and determine in which materialized view the image should be placed. Any subsequent new images are classified with the materialized views updated with the previously classified images. In general, the classification technique consists of two phases – training and testing where in the training phase, the classifier is trained using a number of images whose classes are already known and identified by class labels. Once the training phase is complete, the testing phase is performed in which the now trained classifier is used to determine the class of new, unclassified images i.e. images whose class label is not known. Since classification involves a training phase involving labeled samples, it falls under the category of supervised learning. There are various methods that can be used for performing classification based on supervised learning. The basic method that has been used here in the proposed research work is support vector machine which is a widely used supervised machine learning algorithm.

2. Related Work

Image classification as well as retrieval is possible from an image database, but managing images in the database will be cumbersome if the images are not properly indexed. Content Based Image Retrieval or CBIR is a method of searching images from an image database by querying the images based on some significant features of the concerned images. Hence, to implement the operations required by any CBIR technique, images cannot be directly stored in the image database; instead they need to be stored with proper cataloging on their important features. Some earlier methods in the field of content based image retrieval have been discussed in [5] and [6] and both of the research papers had discussed methods of performing content based image retrieval using color-based, shape-based and depth-based features of the images. In the present research work also, the color-based approach, based on color moments algorithm and shape-based approach, based on height and width ratio have been used to perform image retrieval and image classification. In the same domain, to increase the accuracy as well as the efficiency of the method, a number of critical areas like feature extraction and selection was considered in [7]. Another work on this domain dealt with data representation and similarity measure as was discussed in [8].

Of late, a significant work on CBIR has been done using deep learning techniques [17] which tried to deal with feature representation and similarity measurements in CBIR operation. Another recent research work has utilized some low level features of an image for feature extraction using discrete wavelet transform technique [18]. CBIR technique has also been proposed based on a fusion on visual words with the help of Speeded-Up Robust Features (SURF) and Fast Retina Keypoint (FREAK) features [19]. A method called Incremental Filtering Feature Selection has been proposed in [20] to utilize fuzzy rough set for choosing significant features from images to be incorporated in CBIR technique. A significant work on CBIR technique has been proposed in [21] where multi-motif co-occurrence matrix has been used. Another work has been described in [22] where Particle Swarm Optimization and k-Means Clustering algorithms have been used to design a CBIR system.

There are some commercial products that work with image database, store entire images and for each image queries, entire images are compared for implementing search operation. This way of implementing image query is very much time consuming, especially for the image databases which store large images and the number of images is huge. A considerable improvement on this approach is based on clustering techniques as proposed and discussed in [9] and according to those approaches, similar images are grouped together before the image matching process or any image query operation is initiated. A further improvement on this can be done by using materialized views to store defining features of the images. In this approach, instead of storing full images, only few attributes that may uniquely identify an image are stored in the form of materialized views. Naturally, image retrieval time gets reduced significantly and the size of the database that is used to search for an image gets reduced considerably as well. Though there has been a dearth of research work on the field of CBIR using materialized views, a research work to perform content based image retrieval using materialized views was explored in [10]. The method that was proposed in [10], involved utilizing the materialized views to search for and retrieve images from the image database using high level image features.

3. SUPPORT VECTOR MACHINE

The proposed algorithm as will be discussed in the subsequent section of this research paper mainly depends on the approach followed by a support vector machine. A Support Vector Machine, also known as an SVM, falls under the category of supervised learning category of machine learning algorithms [11]. It classifies data by finding the optimal hyperplane that separates data belonging to one class from data belonging to another class. A data class signifies few attributes which may uniquely identify one particular data set from a collection of data or a universal data set. An optimal hyperplane refers to the hyperplane that has the largest distance between the two data classes. Since an SVM is based on finding a hyperplane between two different data classes, it is a binary classifier, but can be augmented by various methods to classify data into additional number of classes.

4. WORK DESCRIPTION

The main problem related to the research work is to classify the uncategorized images into one of two or three materialized views each containing images belonging to a particular category, i.e., a particular class. So, the proposed method can classify two different images and even three different images. To solve this problem, a classifier has been designed based on the features of a support vector machine which is trained using the images in the materialized views. Once the training phase is complete, the classifier accepts as input the image to be classified and then places it in the correct materialized view. The updated materialized view is then used to retrain the classifier before classifying the next uncategorized image and the process is repeated for the subsequent iterations. Image classification involving two classes and three classes both consist of two major phases – an initialization phase in which the training set of images are read and the training matrix is constructed using the features selected from these images, and a training and classification phase in which the classifier is trained and new images are classified using the trained classifier. So the method consists of the following sub-problems:
Feature extraction, classifier training and classification of new images. Each of these sub-problems is described below:
Feature extraction – This process involves extracting or selecting features from the images that will be used to train the classifier and then to classify new images after training. This was implemented in two different ways, at first; feature selection process that was designed was based on the intensity of the images. Later, the feature extraction process has been modified and instead of considering the intensity of the images, the shapes of the images have been considered and the latter process has given better results. In this research work, the shape of an image has been taken in the form the ratio of its height with its width. In the initial attempt of classifying the images based on their intensities, the color moments algorithm [12] was used to derive the features for each image. The features produced by this algorithm are based on the intensity distribution of the image. At first, the algorithm has converted the input image into the HSV (Hue, Saturation, Value) color space from the RGB (Red, Green, Blue) color space and then has proceeded to calculate the mean pixel intensity, the variance of the pixel intensities and the skewness of the pixel intensities for each channel, i.e., for Hue, Saturation and Value and this has been done for each image. Thus, for each image, this process has generated nine feature values which together have constituted the feature vector of an image. These feature values have been used to classify each image. The pseudocode for the color moments algorithm is given below:

COLOR_MOMENTS ( )

```
{
        Initialization: The feature vector to be zero
        Input: The image whose feature vector is to be calculated
        The input image is converted from the RGB color space to HSV color space
        Extract three individual channels in the form of Hue, Saturation and Value
        For each channel
        Loop
           The mean pixel intensity is calculated
           The variance of the pixel intensity is calculated
           The skewness of the pixel intensity is calculated
        End Loop
        The feature vector is stored with nine values obtained through the above iteration
        The features vector is returned
}
```

Although the classification process by means of the color moments algorithm may produce satisfactory results, the process may not be optimal since it is possible for two classes of images to have similar intensity distributions, particularly if images are in grayscale. Thus, a second approach has been generated based on the shapes of the images. Although two different classes of images may share similar intensity distributions, their shapes should be significantly different in most instances. Thus, generating features based on the shapes of the principal object in the image would be more effective in performing image classification. Using the shape as the principal feature for classification would allow the classifier to be able to differentiate between different classes of images regardless of similar intensity distributions, if any, especially in the case of grayscale images as discussed above. Since the classifier can only work with numerical feature values, a numerical representation for the image shape would be required. Such a representation has been produced by taking the ratio of the height of the image shape to the width of the shape and this ratio has been considered to be the feature for each image. Thus, while the previous method required needed to store a nine-element feature vector for each image, in this method, only one feature value for each image needs to be stored, thus saving disk space as well. Since the image shape is also a better discriminator between images than intensity distribution, the classification accuracy has also been improved, as will be discussed in a later section. The following is the pseudocode for the shape based feature selection process:

EXTRACTION_SHAPE ( )

```
{
   Input: The image whose feature vector is to be calculated
   The input image is converted into a grey-level image
   The points of interest are detected on the converted image
   The most important, i.e., the most useful 100 points are identified on the detected   points
   The coordinates of the selected points are determined
   x ← the x-coordinates of all the points
   y ← the y-coordinates of all the points
   height ← max(y) – min(y) /* max ( ) and min ( ) are tow function to respectively calculate the maximum and the minimum
   values from a list */
   width ← max(x) - min(x)
   ratio ← height / width
   The value stored in ratio is returned
}
```

Classifier training – The classifier is trained using the image features of all the images in the training data set.
Classification of new images – The same features are extracted from the images to be classified in order to classify the images using the previously trained classifier.

The primary difference between the two-class classification and the three-class classification is that for the latter three training matrices are constructed and correspondingly, three different classifiers are trained. The class label for the new image is determined by taking the majority of the labels output by the three classifiers. The algorithm for the process is given below:
Step 1: Images present in the training set are read.
Step 2: The features for each of the images to be trained are calculated.
Step 3: The training matrix using the computed image features and associated class label is computed. The computation is done for each image.
Step 4: For the first iteration, the classifier is trained using the data obtained in the training matrix and for any subsequent iteration, the classifier is retrained with addition featured data, if any.
Step 5: The image to be classified is read.
Step 6: The features of the newly accepted image, as given in step 5 are calculated.
Step 7: The class label of the new image is computed after invoking the classifiers that have been created in step 3.
Step 8: The training matrix is augmented with the features and the class level of this new image. The augmentation is done for the incremental mining of the image features.

Step 9: If any more image is to be classified, then the control is transferred to step 4; otherwise the process is stopped.

In the above mentioned algorithm, the first three steps may be considered to be the initialization phase whereas the next steps may be identified as the training and classification phase. This algorithm has been designed with the help of the characteristics of supervised machine learning approach, specifically, support vector machine as already mentioned in the previous sections. The algorithm, named as IMG_CLASSIFICATION for image classification using the features of an SVM is given below. This algorithm classifies two different classes of images and hence this is dependent on two different materialized views, viz., MV_1 and MV_2. These two views already store the features of the images corresponding to two different classes.

```
IMG_CLASSIFICATION( )
{
  Initialization:  The feature vector to be empty and the list of labels to be empty
  Input: Two materialized views MV1 and MV2
  For each image I in MV1
  Loop
    The required features are selected using the feature selection process
    The features are added to the feature vector
    The label corresponding to MV1 is added to the list of labels
  End Loop
  The above loop iterated for each image in MV2 too.
  The classifier is trained with the feature vector and the list   of labels as parameters
  For each image J to be classified
  Loop
    The required features are selected using the same selection process
    The image J is classified using the selected features
    Image J is added to either MV1 or MV2 depending on the features obtained
    If J is classified to be in MV1
    Then
      MV1 is updated
    Else
      MV2 is updated
    End If
  The classifier is retrained using the newly updated materialized view
  End Loop
}
```

Since a support vector machine is capable of only binary classification, to design an algorithm for more than two-class classification, the method of binary classification has to be used, i.e., a ternary classification has to be performed by means of a binary classifier. Since in a ternary classification, three classes are required, three different SVMs have been trained, one for each of the three pairs of classes possible among the three different classes. Thus, each of these SVMs is a binary classifier as before. Once these three classifiers have been trained, new images having three different types of attributes can be classified. This is done for each image by attempting to classify it using each of the three classifiers. Each classifier would output the class which it has classified the new image into. To determine the correct class of the new image, the class which has been output by a majority of the classifiers has been chosen and in the present method of the ternary classification, a majority means two classes out of three classes. The following example demonstrates this:
Assuming that there are three classes C_1, C_2 and C_3, three binary SVMs have been trained for classifying between C_1 and C_2, between C_2 and C_3 and finally between C_3 and C_1. Now, if an image that should be classified as belonging to class C_1 is taken as an input, each of the three SVMs are executed on the unclassified image. The C_1 versus C_2 classifier would output class C_1, the C_3 versus C_1 classifier will output class C_1 and the C_2 versus C_3 classifier will output either C_2 or C_3 depending on which of these two classes the image is closer to. Since class C_1 is the majority output (i.e., output by 2 of the 3 classifiers) class C_1 is to be taken as the correct class of the image and this is the desired output. The scenario of generating three different outputs for three different binary classifiers is impracticable as explained below:
If for an image, the C_1 versus C_2 classifier outputs C_1 and the C_3 versus C_1 classifier outputs C_3, then there would be no majority output class if the C_2 versus C_3 classifier now outputs C_2. But this is not possible, as the test image is closer to C_1 than to C_2 as determined by the first classifier, and also because it is closer to C_3 than to C_1 as determined by the second classifier, it logically follows that the image must be closer to C_3 than to C_2 and hence the C_2 versus C_3 classifier would definitely output C_3 and thus the resultant class of the new image would be class C_3.
The following is the pseudocode for three-class classification where MV1, MV2 and MV3 are three materialized views containing the required feature of three different image classes:

```
THREE_CLASS_CLASSIFICATION ( )
{
  Initialization: Three feature vectors C1, C2 and C3 to be empty and three lists of labels L1, L2 and L3 to be empty
  For each image I in MV1
```

Loop
The required features are selected using the feature selection process
The features are added to the feature vectors C_1 and C_2
The label corresponding to MV_1 is added to the lists of labels L_1 and L_2
End Loop
For each image I in MV_2
Loop
The required features are selected using the feature selection process
The features are added to the feature vectors C_3 and C_1
The label corresponding to MV_2 is added to the lists of labels L_3 and L_1
End Loop
For each image I in MV_3
Loop
The required features are selected using the feature selection process
The features are added to the feature vectors C_2 and C_3
The label corresponding to MV_3 is added to the lists of labels L_2 and L_3
End Loop
The C_1 versus C_2 classifier is trained with the feature vectors C_1 and C_2 and the lists of labels L_1 and L_2 as inputs
The C_3 versus C_1 classifier is trained with the feature vectors C_3 and C_1 and the lists of labels L_3 and L_1 as inputs
The C_2 versus C_3 classifier is trained with the feature vectors C_2 and C_3 and the lists of labels L_2 and L_3 as inputs
For each image J to be classified
Loop
The required features are selected using the same selection process
The image J is classified using the selected features using each of the three classifiers.
From the three outputs produced by the three classifiers, the output that occurs twice is selected
Image J is added to MV_1, MV_2 or MV_3 depending on the output obtained
The feature vectors of MV_1, MV_2 or MV_3 are updated using the features of the newly classified image
The classifiers are retrained using the newly updated materialized view
End Loop
}

5. Results And Analysis

All the algorithms as mentioned in the previous section have been implemented and the application has been tested with standard data sets. The system that was used to implement the algorithm had Intel Core 2 Duo processor with a speed of 2.0 Ghz. The operating system used was Windows 7 and the software tool used to write the code was MATLAB 2015b. Though the application can directly executed in a system having any version of Windows operating system above the XP version and the runtime environment that is required is MATLAB runtime. The initial testing for the two-class image classifier made use of the color moments feature selection process. To test it, the images of elephants and pandas from a dataset that was accessible at [13] were used.

In the second approach of the image classification as mentioned in the previous section, the method was based on the image shape and accordingly, testing was done with the new process using the same set of images as above and the latter process gave better results. Having successfully tested the classifier on animal images, leaf classification was conducted. The leaf images were taken from the dataset as mentioned in [14]. The leaf images were mainly used for the ternary classifier as proposed in the previous section. To do this, three different types of leaves were taken and to train the classifier, 80% of images under each category were taken. After training the classifier, rest of the images were used for the testing purpose. Since the ternary classifier depends on the binary classifier, to train the former, the latter were automatically tested. So, as far as the binary classifier is concerned, training and testing were conducted for three different pairs of image classes.

With all the above mentioned datasets, the results that were obtained after implementing algorithms as mentioned in the previous section are given next. Table 1 shows only one snapshot of the result containing thirty images obtained after applying COLOR_MOMENTS ( ) with IMG_CLASSIFICATION ( ) and EXTRACTION_SHAPE ( ) with IMG_CLASSIFICATION ( ) on the animal images.

Table 1: Animal Image Classification

Test No.	File Name	Class	COLOR_ MOMENTS( ) and IMG_CLASSIFICATION( )	EXTRACTION_ SHAPE and IMG_CLASSIFICATION( )	
			Classified correctly (Yes/No)?	Height/Width ratio	Classified correctly (Yes/No)?
1	image_0013	Panda	No	1.04	Yes
2	image_0032	Panda	No	0.691	No
3	image_0033	Panda	No	0.632	No

4	image_0035	Panda	No	1.52	Yes
5	image_0036	Panda	Yes	1.07	Yes
6	image_0037	Panda	Yes	0.753	No
7	image_0038	Panda	Yes	1.03	Yes
8	image_0038_2	Elephant	No	0.723	Yes
9	image_0039	Elephant	Yes	0.676	Yes
10	image_0040	Elephant	No	0.614	Yes
11	image_0041	Elephant	Yes	0.807	Yes
12	image_0042	Elephant	Yes	0.877	Yes
13	image_0043	Elephant	Yes	0.723	Yes
14	image_0044	Elephant	Yes	0.814	Yes
15	image_0045	Elephant	Yes	0.661	Yes
16	image_0046	Elephant	No	0.912	Yes
17	image_0047	Elephant	Yes	0.849	Yes
18	image_0048	Elephant	Yes	0.858	Yes
19	image_0049	Elephant	Yes	0.690	Yes
20	image_0050	Elephant	Yes	0.690	Yes
21	image_0051	Elephant	Yes	0.678	Yes
22	image_0052	Elephant	Yes	0.727	Yes
23	image_0053	Elephant	Yes	0.937	Yes
24	image_0054	Elephant	Yes	0.956	Yes
25	image_0055	Elephant	Yes	0.794	Yes
26	image_0056	Elephant	Yes	0.764	Yes
27	image_0057	Elephant	No	1.13	No
28	image_0058	Elephant	Yes	0.646	Yes
29	image_0059	Elephant	Yes	0.871	Yes
30	image_0062	Elephant	No	0.735	Yes

Table 1 shows a comparative study of the outputs obtained when binary classification was implemented after considering the features related to the colors of the images as well as the shapes of the images. The algorithms COLOR_MOMENTS ( ) and IMG_CLASSIFICATION ( ) were applied on the animal images from the dataset obtained from [13] to consider color related features and the algorithms EXTRACTION_SHAPE ( ) and IMG_CLASSIFICATION( ) were used to the shape related features of the images. All the file names have been shown in table 1 are from that dataset. The classification accuracy after considering color related features was 70% whereas the same for the shape related features was 86.7%, which is a reasonable improvement. Figure 1 gives a graphical comparison of the outputs obtained.

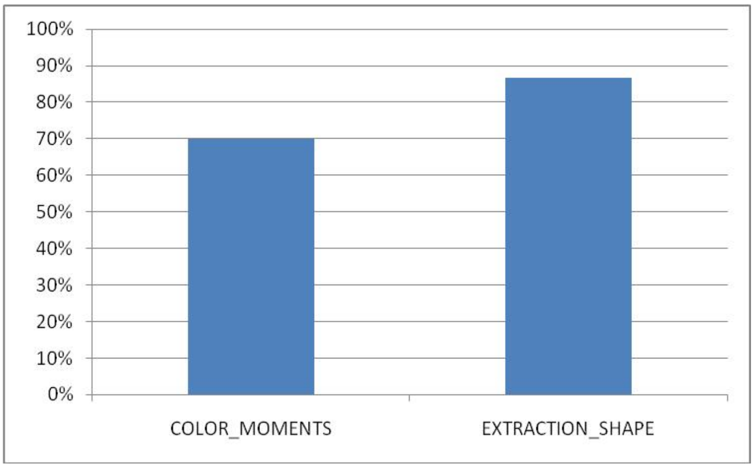

Figure 1: A comparative study of the outputs obtained from two different methods shown in table 1

Table 2 shows a snapshot of the comparative study of the outputs obtained when binary classification was implemented after considering the features related to the colors of the images as well as the shapes of the images. The algorithms COLOR_MOMENTS ( ) and IMG_CLASSIFICATION ( ) were applied on the leaf images from the dataset obtained from [14] to consider color related features and the algorithms EXTRACTION_SHAPE ( ) and IMG_CLASSIFICATION( ) were used to the shape related features of the images.

Table 2: Leaf Image Classification

Test No.	File Name	Class	COLOR_ MOMENTS( ) and IMG_CLASSIFICATION( )	EXTRACTION_ SHAPE and IMG_CLASSIFICATION( )	
			Classified correctly (Yes/No)?	Height/Width ratio	Classified correctly (Yes/No)?
1	1110	Class 1	Yes	0.701	Yes
2	1111	Class 1	Yes	0.707	Yes
3	1112	Class 1	Yes	0.756	Yes
4	1113	Class 1	Yes	0.714	Yes
5	1114	Class 1	Yes	0.768	Yes
6	1115	Class 1	Yes	0.801	Yes
7	1116	Class 1	Yes	0.729	Yes
8	1117	Class 1	Yes	0.731	Yes
9	1118	Class 1	Yes	0.674	Yes
10	1119	Class 1	Yes	0.691	Yes
11	1120	Class 1	Yes	0.681	Yes
12	1121	Class 1	Yes	0.674	Yes
13	1122	Class 1	Yes	0.407	Yes
14	1181	Class 2	Yes	0.891	Yes
15	1182	Class 2	Yes	0.892	Yes
16	1183	Class 2	Yes	0.986	Yes
17	1184	Class 2	Yes	0.967	Yes
18	1185	Class 2	Yes	1.04	Yes
19	1186	Class 2	Yes	0.919	Yes
20	1187	Class 2	Yes	1.06	Yes
21	1188	Class 2	Yes	0.918	Yes
22	1189	Class 2	Yes	0.944	Yes
23	1190	Class 2	Yes	0.960	Yes
24	1191	Class 2	Yes	0.970	Yes
25	1192	Class 2	Yes	0.931	Yes
26	1193	Class 2	Yes	0.970	Yes
27	1194	Class 2	Yes	0.971	Yes

All the file names have been shown in table 2 are from that dataset. Incidentally, the classification accuracies in both the cases here were 100%. So, in this case, no graphical comparison needs to be shown.

Table 3 distinguishes images belonging to one of the three classes – class 1, class 2 and class 3 from the dataset obtained from [14]. So, this table shows an instance of the result obtained after applying ternary classification based on the shapes of the images. As shown in table 3, in the ternary classification process, there is only one case where the classifier has failed to correctly classify the image and the accuracy of the result is 97.37%.

Table 3:Ternary Classification

Test No.	File Name	Class	Height/Width ratio	Classified correctly (Yes/No)?
1	1110	Class 1	0.701	Yes
2	1111	Class 1	0.707	Yes
3	1112	Class 1	0.756	Yes
4	1113	Class 1	0.714	Yes
5	1114	Class 1	0.768	Yes
6	1115	Class 1	0.801	Yes
7	1116	Class 1	0.729	Yes

8	1117	Class 1	0.731	Yes
9	1118	Class 1	0.674	Yes
10	1119	Class 1	0.691	Yes
11	1120	Class 1	0.681	Yes
12	1121	Class 1	0.674	Yes
13	1122	Class 1	0.407	No
14	1181	Class 2	0.891	Yes
15	1182	Class 2	0.892	Yes
16	1183	Class 2	0.986	Yes
17	1184	Class 2	0.967	Yes
18	1185	Class 2	1.04	Yes
19	1186	Class 2	0.919	Yes
20	1187	Class 2	1.06	Yes
21	1188	Class 2	0.918	Yes
22	1189	Class 2	0.944	Yes
23	1190	Class 2	0.960	Yes
24	1191	Class 2	0.970	Yes
25	1192	Class 2	0.931	Yes
26	1193	Class 2	0.970	Yes
27	1194	Class 2	0.971	Yes
28	1251	Class 3	0.501	Yes
29	1252	Class 3	0.554	Yes
30	1253	Class 3	0.446	Yes
31	1254	Class 3	0.450	Yes
32	1255	Class 3	0.536	Yes
33	1256	Class 3	0.456	Yes
34	1257	Class 3	0.476	Yes
35	1258	Class 3	0.508	Yes
36	1259	Class 3	0.454	Yes
37	1260	Class 3	0.546	Yes
38	1261	Class 3	0.528	Yes

In general, by going through the results shown in the above tables, it may be inferred that shape-based classification process performs in a better way than the algorithm based on the color features of the images and the ternary classification process which in turn invokes the binary classifier, gives a satisfactory result. Moreover, as only some defining features against a particular image was required to be stored in the above-mentioned methodologies, there has been a significant improvement in terms of amount of memory space used in storing an image . So for classification and retrieval purpose, an image of size in the order of kilobytes or megabytes has been stored in a memory space whose size is in the order of bytes.

6. Conclusion And Future Scope

The image classifier system based on materialized view as proposed in this paper stores only the defining points or characteristics of the images instead of storing entire images. This has been the major improvement over the existing systems of image classification. This proposed classifier has been able to classify images belonging to three different classes. The number of classes can be increased by incorporating further processing of the binary classifier and the resultant steps will also increase. The classifier gave more accurate result when shape-based features were considered instead of the color-based features of the images. Few constraints are to be maintained in the shape-based feature selection. Among them, one point is the orientation of an image. The classification accuracy may be adversely affected if the orientation of the images changes. Moreover, to reduce the heterogeneity of the image orientation, some established image registration algorithms may be used before actually using the images in the classification of the images. As a future prospect, these things may be taken into consideration for proposing better algorithms to develop image classifiers based on materialized view.

References

1. Aouiche, K., Jouve, P.: Clustering-based materialized view selection in data warehouses. In Proceedings of 10th East European conference on Advances in Databases and Information Systems, 2006, pp. 81 – 95.

2. Docs.Oracle.com: Materialized View Concepts and Architecture, available at https://docs.oracle.com/cd/B10501_01/server.920/a96567/repmview.htm, accessed: 17 – April - 2023.
3. D. Datta and K.N.Dey: Materialized View Generation using Apriori Algorithm. International Journal of Database Management Systems, Volume – 7, Number – 6, 2015, pp. 17 – 27.
4. T. V. Vijay Kumar, Santosh Kumar: Materialized View Selection using Genetic Algorithm. In Proceedings of the 5th International Conference, IC3, 2012, pp. 225 – 237.
5. R. Chaudhuri and A. M. Patil: Content Based Image Retrieval using Color and Shape Features. IJAREEIE, Volume 1, Issue 5, 2012, pp. 386 – 392.
6. A. Jain, R. Muthuganapathy and K. Ramani: Content-Based Image Retrieval using Shape and Depth from an Engineering Database. Proceedings of the 3rd International Conference on Advances in Visual Computing, Volume Part II, 2007, pp. 255 – 264.
7. Zhou H, Yuan Y and Sadka A.H.: Application of semantic features in face recognition. Pattern Recognition, Volume 41, Issue 10, 2008, pp. 3251 – 3256.
8. Eakins J. and Graham M.: Content-based Image Retrieval. A report to the JISC Technology Applications program, University of Northumbria, Newcastle, 1999.
9. Jin J., Kurniawati R., Xu G. and Bai X.: Using browsing to improve content based image retrieval. Journal of Visual Communication and Image Representation, 2001, pp. 123 – 135.
10. M. Ali, A. Adnan, M. Saqib and Zahidullah: Content Based Image Retrieval (CBIR) using Materialized Views. In Proceedings of the International Conference of Computer Science and Information Technology, 2011, pp. 116 – 119.
11. J. Nayek, B. Nayek and H. S. Behera: A Comprehensive Survey on Support Vector Machine in Data Mining Tasks: Applications and Challenges. International Journal of Database Theory and Application, Volume 8, Number 1, 2015, pp. 169 – 186.
12. N. A. C. Hussin, N. Jamil, S. Nordin, K. Awang: Geometrical-Invariant grid-based colour moment (GBCM) for plant identification. Advanced Science Letters, Volume 4, Issue 10 – 11, 2014, pp. 1914 – 1917.
13. L. Fei-Fei, R. Fergus and P. Perona: Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categorie. IEEE.CVPR 2004, Workshop on Generative-Model Based Vision, retrieved from http://www.vision.caltech.edu/datasets/, accessed: 18 – June - 2023.
14. Stephen Gang Wu, Forrest Sheng Bao, Eric You Xu, Yu- Xuan Wang, Yi-Fan Chang and Chiao-Liang Shiang: A Leaf Recognition Algorithm for Plant classification Using Probabilistic Neural Network. IEEE 7th International Symposium on Signal Processing and Information Technology, Cario, Egypt, retrieved from https://sourceforge.net/projects/flavia/, accessed: 03 – November - 2023.
15. S. Agrawal, S. Chaudhuri and V. R. Narasayya: Automated Selection of Materialized Views and Indexes in SQL Databases, In Proceedings of VLDB, 2000, pp. 496 – 505.
16. R.G. Bello, K. Dias, J. Feenan, J. Finnerty, W.D. Norcott, H. Sun, A. Witkowski and M. Ziauddin: Materialized Views in Oracle. In Proceedings of VLDB, 1998, pp. 659 – 664.
17. Rani Saritha, Varghese Paul and P. Ganesh Kumar, "Content based image retrieval using deep learning process", Cluster Computing, Volume 22, March, 2019, pp. 4187 - 4200.
18. Rehan Ashraf, Mudassar Ahmed, Sohail Jabbar, Shehzad Khalid, Awais Ahmad, Sadia Din, Gwangil Jeon, Content Based Image Retrieval by Using Color Descriptor and Discrete Wavelet Transform, Journal of Medical Systems, March, 2018, pp. 42 - 44.
19. Safia Jabeen,Zahid Mehmood,Toqeer Mahmood,Tanzila Saba,Amjad Rehman,Muhammad Tariq Mahmood, An effective content-based image retrieval technique for image visuals representation based on the bag-of-visual-words model, PLOS One, March, 2018, pp. 1 - 24.
20. V. Yadaiah, Dr. R .Vivekanandam and Rajaram Jatothu, A Fuzzy Logic Based Soft Computing Approach in CBIR System Using Incremental Filtering Feature Selection to Identify Patterns, International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 5, 2018, pp. 2432 – 2442.
21. A. Obulesu, V. Vijay Kumar and L. Sumalatha, Content Based Image Retrieval Using Multi-Motif Co-Occurrence Matrix, International Journal of Image, Graphics and Signal Processing, April, 2018, pp. 59 – 72.
22. Shailesh Kumar and Arun Kumar Shukla, Design and Analysis of CBIR System using Hybrid PSO and K-Mean Clustering Methods, International Journal of Current Engineering and Technology, ISSN: 2277 – 4106, Volume 7, Number 2, April, 2017, pp. 397 – 401.