Feature Extraction for Object Recognition and Image Classification

Aastha Tiwari
Banasthali University

Anil Kumar Goswami
DRDO

Mansi Saraswat
Banasthali University

Abstract

Feature Extraction is one of the most popular research areas in the field of image analysis as it is a prime requirement in order to represent an object. An object is represented by a group of features in form of a feature vector. This feature vector is used to recognize objects and classify them. Previous works have proposed various feature extraction techniques to find the feature vector. This paper provides a comprehensive framework of various feature extraction techniques and their use in object recognition and classification. It also provides their comparison. Various techniques have been considered and their pros and cons along with the method of implementation and detailed experimental results have been discussed.

1. Introduction

Feature Extraction (FE) is an important component of every Image Classification and Object Recognition System. Mapping the image pixels into the feature space is known as feature extraction [1]. For automatic identification of the objects from remote sensing data, they are to be associated with certain attributes which characterize them and differentiate them with each other. The similarity between images can be determined through features which are represented as vector [1]. FE is concerned with the extraction of various attributes of an object and thus associate that object with a feature vector that characterize it. FE is the first step to classify an image and identify the objects. The various contents of an image such as color, texture, shape etc. are used to represent and index an image or an object.
Section 2 of the paper provides the literature survey in this area. In the section 3, various FE techniques will be explained and discussed. Section 4 gives the methodology. Section 5 provides overview of experiments performed and results obtained using these FE techniques. Section 6 provides the conclusion.

2. Literature survey

Feature extraction has a long history and a lot of feature extraction algorithms based on color, texture and shape have been proposed. Feature selection is a critical issue in image analysis. In spite of various techniques available in literature, it is still hard to tell which feature is necessary and sufficient to result in a high performance system.

Color is the first and most straightforward visual feature for indexing and retrieval of images .

The first order (mean), the second order (variance) and the third order (skewness) color moments have been proved to be efficient in representing color distribution of images [2]. An approach that lies between subdividing the images and relying on fully segmented images was proposed by Stricker and Dimai[3]. They worked with 5 partially overlapping, fuzzy regions.

The texture is very important cue in region based segmentation of images. Texture features play a very important role in computer vision and pattern recognition [4]. Texture analysis has a long history and texture analysis algorithms range from using random field models to multiresolution filtering techniques such as the wavelet transform [5]. Due to resemblance between multi-resolution filtering techniques and human visual process, Gabor and Wavelet Transform techniques are often used for texture characterization through the analysis of spatial frequency content [6].

The first two approaches have been explored more thoroughly than shape based approaches. Shape representation and description is a difficult task. This is because when a 3-D real world object is projected onto a 2-D image plane, one dimension of the object Information is lost [7].

ISSN: 2278-0181
Vol. 2 Issue 10, October - 2013

3. Feature extraction

There are various types of feature extraction with respect to satellite images. The similar features together form a feature vector to identify and classify an object. Various feature extraction techniques have been explained in detail

3.1 Color

Color is one of the most important features with the help of which humans can easily recognize images. It is most expressive of all the visual features. It is easy to extract, analyze and represent an object. Due to their little semantic meaning and its compact representation, color features tend to be more domain independent compared to other features [8]. Its property of invariance with respect to the size of the image and orientation of objects on it make it a suitable choice for feature extraction in images. The quality of feature vector depends largely on the color space used for representation. Color features are represented using color moments, fuzzy color moments, color histogram etc. Therefore, it is more suitable for image retrieval.

3.1.1 Color moments. Color distribution of images can be represented effectively and efficiently using color moments. Color moments offer computational simplicity, speedy retrieval, and minimal storage [8]. These are very robust to complex background and independent of image size and orientation [9]**.** Color moments feature vector is a very compact representation as compared to other techniques due to which it may also have lower discrimination power. Therefore, it can be used as the first pass to reduce the search space.

There are first order (mean), second order (variance) and third order (skewness) color moments which are represented as below.

$$Mean = \frac{1}{N*M}\sum_{i=0}^{N-1}\sum_{j=0}^{M-1} P[i][j]$$

$$Variance = \sqrt{\frac{1}{N*M}\sum_{i=0}^{N-1}\sum_{j=0}^{M-1} (P[i][j]-Mean)^2}$$

$$Skewness = \sqrt[3]{\frac{1}{N*M}\sum_{i=0}^{N-1}\sum_{j=0}^{M-1} (P[i][j]-Mean)^3}$$

Where M and N are the image template's height and width and P[i][j] are the pixel values.

All the three orders can be calculated either for each color band of image separately or for gray band. If it is used separately for each of color bands, then there will be *3*p* color moments making a feature vector.

Feature Vector = (Mean$_1$*, Variance*$_1$*, Skewness*$_1$*, Mean*$_2$*, Variance*$_2$*, Skewness*$_2$*, ... Mean*$_p$*, Variance*$_p$*, Skewness*$_p$*)*

where *p* is the number of color bands in an image. In the case of gray image, there will be only 3 color moments. It depends on the application whether to have color moments for each band or for a single band i.e. gray band. As the size of feature vector increases, the need of computational power also increases.

3.1.2 Fuzzy color moments. This technique is an improvement over the previous technique of color moments as it takes into consideration the spatial layout of pixels. The image is partitioned into fuzzy regions i.e. central ellipsoidal region and four surrounding regions, defined by a membership value as shown below.

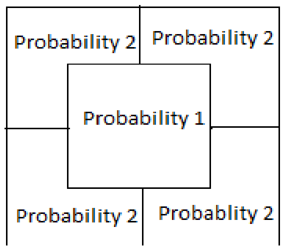

Figure 1. Membership matrix

According to the membership matrix, the pixels located at the centre of the image contribute to the feature vector of the central region only. The pixels located on the border region have a lesser influence. The color moment equations are applied to each fuzzy region and result is obtained as under:

$$F(v) = \sum_{i=1}^{5} u_i * f_i(v)$$

Where F (v) is the overall parameter for v (mean, variance and skewness). u_i is the membership value of a fuzzy region in an image and f_i (v) is the value of v in the i^{th} region.

There is a drawback of this method that if an object exists in the center of a query image, other images containing a similar object not located in the center will not be retrieved. This approach is computationally complex as compared to color moments but it increases the accuracy of the results.

3.1.3 Color histogram. The color histogram approach works on the frequency of occurring of a pixel value in an image. It finds the total number of pixels in each bin which lies in its range. If there are more number of bins in the histogram, more discriminating power it has. However, a large number of bins increases the computation cost.

Color histogram is easy to compute and effectively represents the distribution of pixel colors in image. It takes relatively less time as compared to the classical color moments and fuzzy color techniques. Color histogram has the advantages of speediness and not sensitive to images' changes, such as translation, rotation, scaling etc [9]. The problem with this approach is that it does not take spatial information of pixels into account. It has been shown that moment based color distribution features can be matched more robustly than color histograms as histograms do not capture spatial relationship of color regions and thus, they have limited discriminating power [9].

A solution to this problem can be dividing the image into sub areas and finding the histogram for each of them. This increases the information about location but also increases the memory requirement and computational cost.

A histogram can be represented as-

BANDS / BINS	Band [0]	Band [1]	Band [2]	.....	.....	Band [n]
Bin [0]	$X_{o,o}$	$X_{o,1}$	$X_{o,2}$	.....	.....	$X_{o,n}$
Bin [1]	$X_{1,0}$	$X_{1,1}$	$X_{1,2}$	.....	.....	$X_{1,n}$
Bin [2]	$X_{2,0}$	$X_{2,1}$	$X_{2,2}$	.....	.....	$X_{2,n}$
.....	.....	.....	.....	.....	.....	.....
.....	.....	.....	.....	.....	.....	.....
Bin [m]	$X_{m,0}$	$X_{m,1}$	$X_{m,2}$	.....	.....	$X_{m,n}$

Where $X_{i,j}$ is the number of pixels in i^{th} bin and j^{th} band

Figure 2. Color histogram

3.2 Texture

Texture is one of the important features in image analysis for many applications. Texture analysis attempts to quantify intuitive qualities described by terms such as rough, smooth, silky, or bumpy as a function of the spatial variation in pixel intensities. The choice of the textural features should be as compact as possible and yet as discriminating as possible [14]. It is essential to find a set of texture features with good discriminating power, in order to design an efficient algorithm for texture classification.

Texture features can be found using methods as Gabor Filter, Haar Wavelet Decomposition and Wavelet GLCM etc.

3.2.1 Haar wavelet. The Haar Wavelet Transform (HWT) is one of the simplest and basic transformations from the space domain to a local frequency domain [10]. It is composed of a sequence of low-pass and high-pass filters, known as a filter bank. Haar wavelets are the fastest to compute and have been found to perform well in practice. The HWT is a discrete wavelet transform and is therefore preferred since it provides temporal resolution i.e. it captures both frequency and spatial information. It decomposes the image into three detailed sub bands and an approximation image which can be decomposed further [11]. HWT enable us to speed up the wavelet computation phase for thousands of sliding windows of varying sizes in an image. HWT facilitates the development of efficient incremental algorithms for computing wavelet transforms for larger windows in terms of the ones for smaller windows. One disadvantage of Haar wavelets is that it tends to produce large number of signatures for all windows in image [10]. The other disadvantage of the Haar wavelet is that it is not continuous, and therefore not differentiable. This property can, however, be an advantage for the analysis of signals with sudden transitions, such as monitoring of tool failure in machines [12]. The figure below shows the decomposition of image matrix at each level.

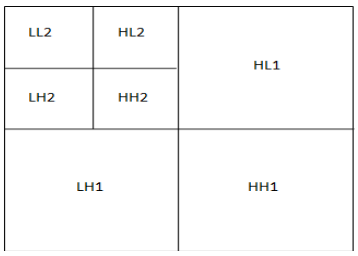

Figure 3. Two level wavelet decomposition

Let us consider two 2×2 matrices: H and Y.

If

$$H = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

Then,

$$Y = \begin{pmatrix} a+b+c+d & a-b+c-d \\ a+b-c-d & a-b-c+d \end{pmatrix}$$

where Y represents the Haar Transform of H.

3.3 Shape

Shape is one of the most important features in feature extraction. They are usually described when the image has been segmented into different regions or objects. Shape description can be categorized into either region based or boundary based. A good shape representation feature for an object should be invariant to translation, rotation and scaling.

3.3.1 Shape moment invariant. Moment invariants are useful features of a two-dimensional image as they are invariant to shifts, to changes of scale and to rotations, or to shifts and to general linear transformations of the image [2]. The results show that recognition schemes based on shape moment invariants could be truly position, size and orientation independent, and also flexible enough to learn almost any set of patterns. This method can be generalized to accomplish pattern identification not only independently of position, size and orientation but also independently of parallel projection [13]. Even if they suffer from some limitations, they frequently serve as a reference method for evaluation of the performance of other shape descriptors.

If we represent object R as an image, the central moments of the order p + q for the shape of R are defined as:

$$\mu_{p,q} = \sum_{(x,y \in R)} (x - x_c)^p (y - y_c)^q \text{ ; where } x_c, y_c \text{ is the centre of the object.}$$

This central moment can be normalized to be scale invariant.

$$\eta_{p,q} = \frac{\mu_{p,q}}{\mu_{0,0}^{\gamma}}, \qquad \gamma = \frac{p+q+2}{2}$$

Based on these moments, a set of moment invariants to translation, rotation and scale can be derived[6].

$$\emptyset_1 = \mu_{2,0} + \mu_{0,2}$$
$$\emptyset_2 = (\mu_{2,0} - \mu_{0,2})^2 + 4\mu_{1,1}^2$$
$$\emptyset_3 = (\mu_{3,0} - 3\mu_{1,2})^2 + (\mu_{0,3} - 3\mu_{2,1})^2$$
$$\emptyset_4 = (\mu_{3,0} + \mu_{1,2})^2 + (\mu_{0,3} + \mu_{2,1})^2$$
$$\emptyset_5 = (\mu_{3,0} - 3\mu_{1,2})(\mu_{3,0} + \mu_{1,2})[(\mu_{3,0} + \mu_{1,2})^2 - 3(\mu_{0,3} + \mu_{2,1})^2] + (\mu_{0,3} - 3\mu_{2,1})(\mu_{0,3} + \mu_{2,1})((\mu_{0,3} + \mu_{2,1})^2 - 3(\mu_{3,0} + \mu_{1,2})^2)$$
$$\emptyset_6 = (\mu_{2,0} - \mu_{0,2})((\mu_{3,0} + \mu_{1,2})^2 - (\mu_{0,3} + \mu_{2,1})^2) + 4\mu_{1,1}(\mu_{3,0} + \mu_{1,2})(\mu_{0,3} + \mu_{2,1})$$
$$\emptyset_7 = (3\mu_{2,1} - \mu_{0,3})(\mu_{3,0} + \mu_{1,2})((\mu_{3,0} + \mu_{1,2})^2 - 3(\mu_{0,3} + \mu_{2,1})^2)$$

4. Methodology

The proposed work consists of extracting features from each template of the input image and highlighting the areas similar to the query template. A template is selected as the query template. Its features are calculated and stored in a feature vector. The entire image is divided into various templates of fixed size. The template moves by a fixed pixel distance at each computation. Features of each template are extracted and stored. They are compared with the query features and results are displayed based on the top 10 matches or the threshold value. In the end the same image is displayed highlighting the areas similar to the query template. The block diagram given below shows the entire process :-

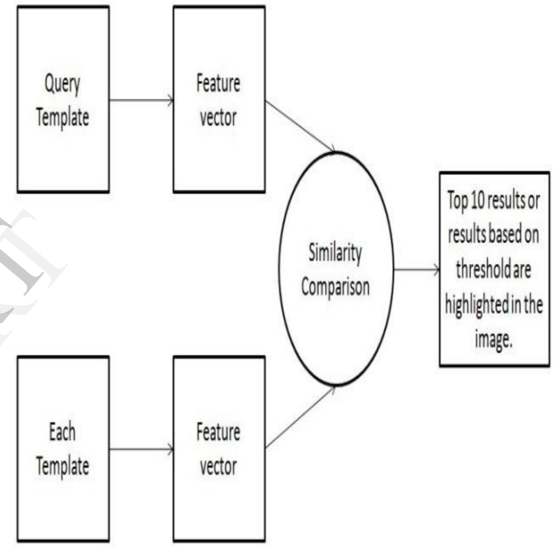

Figure 4. Block diagram of the system

There are various techniques based on color, shape and texture which can be used for feature extraction. Each technique gives a different set of features. The color moments for color extraction gives mean, variance and skewness for each color band. The fuzzy color moments also gives the same set but keeping in consideration the spatial location of pixels. The color histogram gives the number of pixels in each bin in each band. The Haar wavelet method returns the mean and standard deviation of each of the four sub-bands at every decomposition level. Shape moment invariants technique is used for shape extraction which gives a set of seven moment invariants.

The techniques used for feature extraction based on color, texture or shape can be implemented as below:-

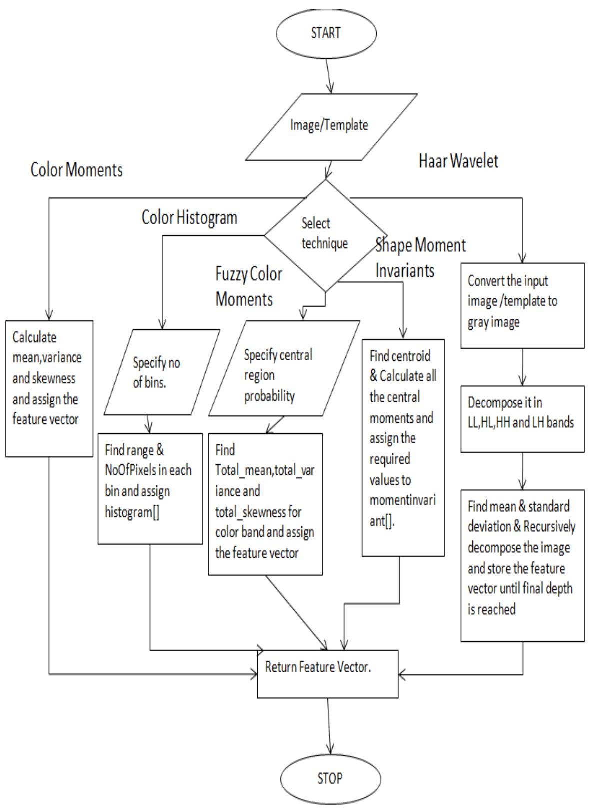

Figure 5. Flowchart showing the implementation details

The size of feature vector depends upon the technique selected. The table below shows the size of feature vector obtained by the techniques described in this paper:-

Table 1. Size of feature vector

S.NO	TECHNIQUE	SIZE OF FEATURE VECTOR
1	Color Moments	No Of Bands*3
2	Fuzzy Color Moments	No Of Bands*3
3	Color Histogram	No Of Bands*No Of Bins
4	Shape Moment Invariants	7
5	Haar Wavelet	2*(Depth*3+1)

The algorithms for implementing various techniques are as follows:-

4.1 Color moments

1. Inputs –
   a) ImageTemplateHeight
   b) ImageTemplateWidth
   c) NoOfBands
   d) ImageTemplateDataMatrix[Height][Width][NoOfBands]
2. Find Mean, Variance and Skewness for each band
   a) for k->0 to NoOfBands-1
      for i->0 to ImageTemplateHeight-1
      for j->0 to ImageTemplateWidth-1
      sum+=ImageTemplateDataMatrix[i][j][k]
      Mean(k)=sum/(ImageTemplateHeight*ImageTemplateWidth)
   b) for k->0 to NoOfBands-1
      for i->0 to ImageTemplateHeight-1
      for j->0 to ImageTemplateWidth-1
      sum1+=(ImageTemplateDataMatrix[i][j][k]-Mean)^2
      sum2+=(ImageTemplateDataMatrix[i][j][k]-Mean)^3
      Variance(k)=[sum1/(ImageTemplateHeight*ImageTemplateWidth)]^1/2
      Skewness(k)=[sum2/(ImageTemplateHeight* ImageTemplateWidth)]^1/3
   c) for k->0 to NoOfBands-1
      FeatureVector[k]=Mean(k)
      FeatureVector[k]=Variance(k)
      FeatureVector[k]=Skewness (k)
3. Output - FeatureVector

4.2 Fuzzy color moments

1. Input
   a) ImageTemplateHeight
   b) ImageTemplateWidth
   c) NoOfBands
   d) ImageTemplateDataMatrix[Height][Width][NoOfBands]
   e) Central Region Probability
2. Divide the image into 5 regions for each color band
   a) for k->0 to NoOfBands-1
      for i->0 to ImageTemplateHeight-1
      for j->0 to ImageTemplateWidth-1
      i. sum[region no]+=ImageTemplateDataMatrix[i][j][k]
      ii. NoOfPixels[region no]++
   Find Mean
      for c->0 to NoOfRegions-1
      Mean(c)=sum[c]/ NoOfPixels[c]
   b) Find variance and skewness for each region
      for k->0 to NoOfBands-1
      for i->0 to ImageTemplateHeight-1
      for j->0 to ImageTemplateWidth-1
      sum1[region no]+ = (ImageTemplateDataMatrix[i][j][k]-Mean)^2
      sum2[region no]+ = (ImageTemplateDataMatrix[i][j][k]- Mean)^3
      Variance(k)=[sum1/(ImageTemplateHeight*ImageTemplateWidth)]^1/2
      Skewness(k)=[sum2/(ImageTemplateHeight*ImageTemplateWidth)]^1/3

c) Find Total Mean, variance and skewness
for c->0 to NoOfRegions
TotalMean+=
mean[c]*RegionProbability
TotalSkewness+=
Skewness[c]*RegionProbability
TotalVariance+=
Variance[c]*RegionProbability
d) Feature Vector[]=TotalMean
Feature Vector[]=TotalVariance
Feature Vector[]=TotalSkewness
3. Output - Feature Vector

4.3 Color histogram

1. Input
   a. ImageTemplateHeight
   b. ImageTemplateWidth
   c. NoOfBands
   d. ImageTemplateDataMatrix[Height][Width][NoOfBands]
   e. Number Of Bins
2. Range=256/No Of Bins
3. Row=0
4. For each band
   a. for k->0 to NoOfBands-1
   i. low=0
   ii. high=1
   iii. while (range*high<=256)
   for i=0 to ImageTemplateHeight-1
   for j=0 to ImageTemplateWidth-1
   a. if (range*low<= ImageTemplateDataMatrix[i][j][k] && ImageTemplateDataMatrix[i][j][k]< range*high)
   Histogram[row]++
   b. low++
   c. high++
   d. row++
5. Output Histogram

4.4 Shape moment invariants

1. Inputs –
   a) ImageTemplateHeight
   b) ImageTemplateWidth
   c) NoOfBands
   d) ImageTemplateDataMatrix[Height][Width][NoOfBands]
2. Find the Cartesian product
   for i->0 to Height-1
   for j->0 to Width-1
   i.)sum=sum + i^p* j^q *ImageTemplateDataMatrix[i][j][NoOfBand]
   ii.)C(p, q)=sum
   iii.) return C
3. Find Centroid for each band
   xcentre[band no]=C(1,0)/C(0,0)
   ycentre[band no]=C(0,1)/C(0,0)
4. Central Moment
   for i->0 to Height-1
   for j->0 to Width-1
   CM(p,q)=(j-xcentre[band no])^p*(i-ycentre[bandno])^q*ImageTemplateDataMatrix[i][j][band no]
5. Normalized Central Moment
   NCM(p,q)=CM(p,q)/(CM(0,0)^gamma)
6. Calculate Moment invariants for each band
   for k->0 to No of bands-1
   MomentInvariant[]=NCM(2,0)+NCM(0,2)
   MomentInvariant[]=[NCM(2,0)-NCM(0,2)]^2 +4*NCM(1,1)^2
   MomentInvariant[]=[NCM(3,0)-NCM(1,2)] ^2 +[NCM(0,3)-3*NCM(2,1)]^2
   MomentInvariant[]=[NCM(3,0)+NCM(1,2)]^2 +[NCM(0,3)+NCM(2,1)]^2
   MomentInvariant[]=[3*NCM(3,0)3*NCM(1,2)][NCM(3,0)+NCM(1,2)][NCM(3,0)+NCM(1,2)]^2
   3*(NCM(2,1)+NCM(0,3))^2]+(3*NCM(2,1)-NCM(0,3))(NCM(2,1)+NCM(0,3))*[3*(NCM(3,0)+NCM(1,2))^2-(NCM(2,1)+ NCM(0,3))^2]
   MomentInvariant[]=(NCM(2,0)-NCM(0,2))[(NCM(3,0)+NCM(1,2))^2-(NCM(2,1) +NCM(0,3))^2]+4*NCM(1,1)* (NCM (3,0)+ NCM(1,2))(NCM(2,1)+NCM(0,3)]
   MomentInvariant[]=(3*NCM(2,1)-NCM(0,3))(NCM(3,0)+NCM(1,2))[(NCM(3,0)+NCM(1,2))^2-3*(NCM(2,1)+NCM(0,3))^2]+(3*NCM(1,2)-NCM(3,0)(NCM(2,1)+NCM(0,3))*[3*(NCM(3,0)+NCM(1,2)^2-(NCM(2,1)+NCM(3,0))^2]
7. Output-MomentInvariant

4.5 Haar wavelet transform

1. Inputs –
   a. ImageTemplateHeight
   b. ImageTemplateWidth
   c. NoOfBands
   d. ImageTemplateDataMatrix[Height][Width][NoOfBands]
   e. Depth of Partition

2. Convert the image template to gray Image
   for i->0 to Height-1
   for j->0 to Width-1
   Gray Image[i][j]=
(0.30*ImageTemplateDataMatrix[i][j][0]
+0.59*ImageTemplateDataMatrix[i][j][1]
+0.11*ImageTemplateDataMatrix[i][j][2])
3. Check for the correct depth of partition and assign it maximum possible value if it is more than maximum
4. Calculate Transform
   i.)If Height or width is odd, make it even by subtracting 1.

ii.) Decompose the image into LL,HL,HH and LH band.
iii.)Normalize the value of transformed matrix
iv.)if partitions are less than the depth of partition
   a. calculate mean and standard deviation for the LH,HL and HH band
   b. Store corresponding values in feature_vector
   c. Height=Height/2
   d. Width=Width/2
   e. Calculate Transform for LL band (step 4)
v.)else
   calculate mean and standard deviation for all the 4 bands
5. Output feature_vector

5. Results and discussion

All the above mentioned techniques were applied on 2 sample images Figure 6 and Figure 7 as shown in the table below:

Figure 6. Test image 1 showing the query window

Figure 7. Test image 2 showing the query window

Table 2. Test Data

CATEGORY	TEST IMAGE 1	TEST IMAGE 2
Image Width	500	500
Image Height	500	377
Query Template Width	100	60
Query Template Height	100	60
No Of Results	Top 50	Top 10

The system had 64-bit operating system with 4 GB RAM and Intel(R) Xeon(R) E5504 @ 2.00GHz 2.00GHz

The table given below shows the time taken by each technique and the results obtained in each case have been shown separately.

Table 3. Time taken by each technique

S.NO.	TECHNIQUE	TIME TAKEN (HH:MM:SS:MS)	
		Figure 1	**Figure 2**
1	Color Moments	00:01:18:047	00:00:01:700
2	Fuzzy Color Moments	00:02:03:677	00:00:02:215
3	Color Histogram	00:00:08:721	00:00:00:234
4	Moment Invariant	00:32:31:488	00:00:39:546
5	Haar Wavelet	00:00:07:707	00:00:00:203

The results obtained when different techniques were applied are as shown below:

Color moments:

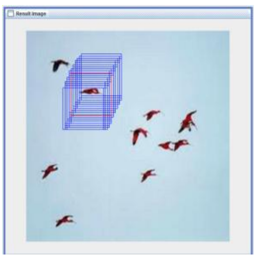

Figure 8

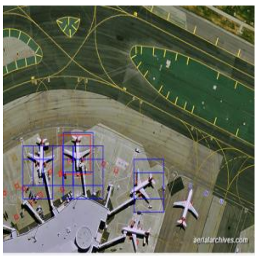

Figure 9

Fuzzy color moments:

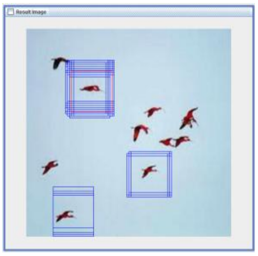
Figure 10

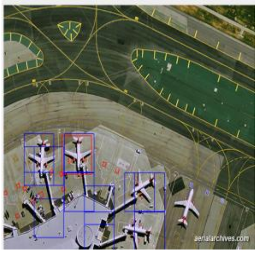
Figure 11

Color histogram:

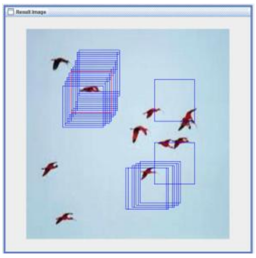
Figure 12

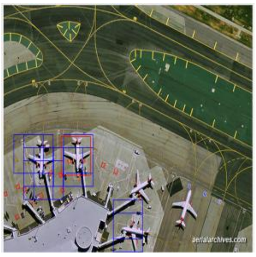
Figure 13

Moment invariants:

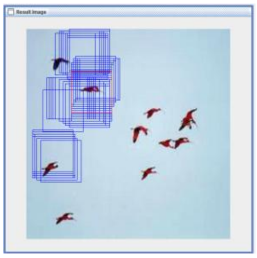
Figure 14


Figure 15

Haar wavelet:

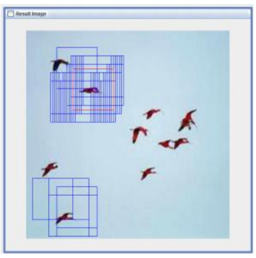
Figure 16

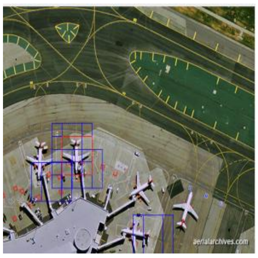
Figure 17

The input image in each case was Figure 6 and Figure 7. The query template in the above images is shown by a red window and the extracted similar areas are highlighted by a blue window. Top 50 results were obtained and highlighted for Figure 6 and top 10 for Figure 7. The results for color moments are given by Figure 8 and Figure 9, fuzzy color moments by Figure 10 and Figure 11 and that for histogram technique by Figure 12 and Figure 13. Figure 14 and Figure 15 show the results for Shape Moment Invariants. The results for Haar wavelet are given by the Figure 16 and Figure 17. As shown by Figure 8 and Figure 9 , the color moment technique gives the best and accurate results. However the time for computation is least in color histogram technique. Fuzzy color moments technique takes the maximum time as compared to other color extraction algorithms, but suits the most for object recognition.

The time complexity for Haar wavelet decomposition is very less. The top 45 results were most precise highlighting the area exactly similar to the query window for Figure 6.

The shape moment invariants take considerable time for execution but gives good results. Most of the highlighted windows were able to recognize the objects similar to the one enclosed by the query window.

6. Conclusion

Various feature extraction techniques have been discussed and compared in the paper. There are trade-offs between various techniques. Some techniques increase the accuracy as compared to other techniques but their time complexity is more. While other techniques provide acceptable results by doing computation relatively fast. Some techniques are storage efficient while others are time efficient. Hence different techniques suit different needs. The accuracy of results also depends upon the size of the window and spatial distribution of pixels. The results obtained show that the Fuzzy Color Moments is comparatively good technique among techniques explained above for object recognition. The work can be further extended by fusion of two or more techniques at the feature level to get better results. This fusion of these techniques is recommendable as it increases the accuracy. Although, it will increase the accuracy but will also increase the computational load. The use of particular technique or the fusion of various techniques depends on the applications for which these are being used.

These various techniques are being used in different applications and areas such as Object Recognition, Image Classification, Content Based Image Retrieval (CBIR), Robotics, and Artificial Intelligence based System, Knowledge Based System or Expert System, Computer Vision, Learning System etc. The framework discussed in this paper will help to develop these systems further.

References

[1] Bhuravarjula, Hari Hara Pavan Kumar, and VNS Vijaya Kumar. "A Novel Image Retrieval Method using color Moments."*International Journal Of Electronics And Computer Science Engineering (Ijecse, ISSN: 2277-1956)* 1.04 (2012): 2432-2438.

[2] Reiss, Thomas H. "The revised fundamental theorem of moment invariants." *Pattern Analysis and Machine Intelligence,IEEE Transactions on* 13.8 (1991): 830-834

[3] Markus Stricker and Alexander Dimai, **"**Color Indexing with Weak Spatial Constraints", *SPIE conference*, San Jose , Feb.96

[4] Vibha S. Vyas and Priti Rege, "Automated Texture Analysis with Gabor Filter",*GVIP journal,*Volume 6, issue 1, July 2006

[5] Manjunath, Bangalore S., and Wei-Ying Ma. "Texture features for browsing and retrieval of image data." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 18.8 (1996): 837-842.

[6] Junyu Dong, Muwei jian, Dawei Gao, Shengke Wang, **"**Reducing the Dimensionality of Feature Vectors Texture Image Retrieval Based on Wavelet Decomposition", *Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*

[7]S. Arivazhagan, L. Ganesan, S.Selvanidhyananthan, "Image Retreival using Shape Feature", *International Journal Of Imaging Science and Engineering(IJISE),* Volume 1, Number 3, July 2007

[8] Banu, M. Sheerin, and Krishnan Nallaperumal. "Analysis of color feature extraction techniques for pathology image retrieval system." *Computational Intelligence and Computing Research (ICCIC), 2010 IEEE International Conference on*. IEEE, 2010.

[9] Huang, Zhi-Chun, et al. "Content-based image retrieval using color moment and Gabor texture feature." *Machine Learning and Cybernetics, (ICMLC), 2010 International Conference on*. Vol. 2. IEEE, 2010.

[10] Pasnur, M. A., and P. S. Malge. "Image Retrieval Using Modified Haar Wavelet Transform and K Means Clustering" *International Journal of Emerging Technology and Advanced Engineering,* Volume 3, Issue 3, March 2013

[11] D N Verma, Sahil Narang, Bhawna Juneja, "Texture based Image Retrieval Using Correlation on Haar Wavelet Transform", Janahan Lal Stephen (Ed.): CNC 2012, LNICST 2012, pp. 81–86. © Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2012

[12] Kekre, H. B., Sudeep D. Thepade, and Akshay Maloo. "Performance Comparison of Image Retrieval Using Fractional Coefficients of Transformed Image Using DCT, Walsh, Haar and Kekre's Transform." *CSC-International Journal of Image processing (IJIP)* 4.2 (2010): 142-155.

[13] Hu, Ming-Kuei. "Visual pattern recognition by moment invariants." *Information Theory, IRE Transactions on* 8.2 (1962): 179-187.

[14] Reddy, BV Ramana, et al. "Classification of Textures Based on Features Extracted from Preprocessing Images on Random Windows." *jiP* 1 (2009): 0.