An efficient framework for image retrieval using color, texture and edge features

J COMPONENT PROJECT REPORT

Review - I

submitted by

Kaustubh Deokar -16BCE0353

Harshit Agarwal-16BCE077

in partial fulfilment for the award of the degree of

B. Tech

in

Computer Science and Engineering

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Vellore-632014, Tamil Nadu, India

School of Computer Science and Engineering

August, 2018

Abstract:

Here in this project we construct an image retrieval system based on three main factors. The proposed framework initially selects pertinent images from a large database using color moment information. Subsequently, Local Binary Pattern (LBP) and Canny edge detection methods are used to extract the texture and edge features respectively, from the query and resultant images of the initial stage of this framework. Then, we calculate the Chi Square distance between the red green and the blue colour channels of the query and the main image. Then these two (the LBP pattern and the edge feature extracted from the canny edge detection and by Chi square method) information about these two features corresponding to the query and selected images are calculated and combined, are then sorted and the nearest 'n' images are presented.

Here we have used two datasets Wang and Corel database in our project. Results shown here in the results section belong to the Wang dataset. The wang dataset contain 1000 images and Corel contain 10000 images.

Introduction:

Digital imaging is an indispensable segment in a multitude of applications such as biomedical imaging, remote sensors, crime detection, education, multimedia, data mining, etc. These applications require digital images as a source for various processes like segmentation, object recognition, tracking, and others. To index and search suitable images from the rapidly increasing digital image collections, an image retrieval system is used. In image databases, for fast and efficient retrieval, images are indexed and searched in two ways, namely, using keywords and using visual contents of an image. The traditional method such as the text-based image retrieval (TBIR) system uses keywords for semantic image retrieval. But this system fails to handle large database because of its non-automatic keyword generation scheme and it fully relies on the perception of human experts who are employed in the keyword generation task. This often leads to inappropriate keyword generation for an image.

Whereas, in our project the Content-Based Image Retrieval (CBIR) system, the visual contents i.e., color, texture, shape, etc., of an image are utilized to give plausible results from the large database. Among the visual contents, color plays a vital role in human perception by giving a pleasant view of the environment. It can also be used to identify an object and distinguish one object from another. Hence, in CBIR, color can be considered as the leading descriptor due to its simplicity in calculation and invariant behavior towards translation, rotation and change in the viewing angle.

Literature Review

Yue et al. [2] uses Color histogram and Gray Level Co-occurrence Matrix (GLCM) features to give more accurate retrieval results in CBIR system. Global and local histograms are evaluated over HSV color space image. Local histogram provides better performance than global histogram and then GLCM is used to extract the texture feature of a gray level image. Local histogram and GLCM features are fused by giving equal weight to color and texture features.

Color Co-occurrence Matrix (CCM), Difference Between Pixels of Scan Pattern (DBPSP) and Color Histogram K-Mean (CHKM) image features are fused [3] to get the highly similar image results. Among the three features, two features (CCM and DBPSP) are used to extract the texture information and the third feature (CHKM) is used to extract only the color information.

Multi-scale edge field method for multimedia retrieval [4] uses Canny edge extraction as a part of process to obtain the object boundaries in different scales. Agarwal et al.

[5] have applied Canny edge detection on the luminance channel of the YCbCr color image in order to improve performance of the image retrieval system.

Liu and Yang [6] have proposed Color Difference Histogram (CDH) on Lab color space that is completely different from long established color histogram method. Lab color space is preferred for estimating CDH because it uses the color difference between color and edge orientation texture details of the image. Subsequently, Canberra distance metric is used to measure the similarity between query and database images. Furthermore, texture and color features-based retrieval is obtained by local extrema peak valley pattern and RGB color histogram [7]. The texture and shape features are mined using Local Ternary Pattern (LTP) and geometric moments to pull off large number of relevant images [8].

Neural network structure [9] is proposed to reduce the semantic gap in image retrieval. The network is trained from the Bags of Images (BOI) which have the third level decomposed wavelet packet tree information and the mean eigen- vector of each Gabor filter response image. Then, Pearson correlation co-efficient is used to find the similarity between the feature vectors. Finally, the outputs are refined with the help of relevance feedback mechanism. But, the training phase complexity and convergence time of this approach are high.

Wang et al. [10] integrates Color, texture and shape features to give efficient retrieval by CBIR system. Color feature is calculated via slightly modified Dominant Color Descriptor (DCD). Here, the image is segmented into eight coarse parts, and then mean value of each segment acts as its quantized color. Next, the clustering is used to merge the nearest colors. Finally, five dominant colors and its percentages are obtained. The texture feature is extracted by taking convolution between image and band pass filter with four directions.

Methodology

Color descriptor

Our first step in the project is to extract the color features of the images in Wang dataset or any dataset that we are using. The query image also needs the color features extracted from it and then calculating the mean and the standard deviation we set a threshold for a high and low to segregate images from the wang dataset.

Here color moments (statistical measure) are chosen to represent the color details of the image It gives the pixel distribution information of the image in two compact forms. The first order moment gives average information about the pixel distribution of a given image. The closeness of the pixel distribution about mean color is estimated by second order moment.

In the first stage of the retrieval process, average color information (mean) and the quantity of the number of pixels that differs from the mean (standard deviation) of the query image are estimated globally from the three-color channels (Red, Green, and Blue) of the RGB color space as follows.

$$Mean(Ic) = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} P_{cij}, \quad c = \{R, G, B\} \qquad Std(Ic) = \left(\frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} \left(P_{cij} - Mean(Ic)\right)^2 \right)^{\frac{1}{2}}, \quad c = \{R, G, B\}$$

The mean value of color image gives the average color information of the image. standard deviation of the image is also important to give details about the distribution of image pixel around the average information.

The code snippet for the same is shown below.

```
meanEachImage=arrayfun(@(x) mean(reshape(imread(cell2mat(fullfile('wangsame',F1(x)))),[],nl
query = imread('query.jpg');
meanofquery =int16(mean2(query));
sdofquery = int16(std2(query));
%lc and hc
lowrange=meanofquery-sdofquery;
highrange=meanofquery+sdofquery;
array1=int16([]);



for i=1:1000
    if(lowrange<cell2mat(meanEachImage(i)) && cell2mat(meanEachImage(i))<highrange)
        array1=[array1,meanEachImage(i),i];
```

Texture descriptor

Texture formally has no definition. Intuitively texture measures the smoothness, coarseness, regularity. The local binary pattern (LBP) texture analysis operator is defined as a gray-scale invariant texture measure, derived from a general definition of texture in a local neighborhood. The current form of the LBP operator is quite different from its basic version: the original definition is extended to arbitrary circular neighborhoods, and a number of extensions have been developed. The basic idea is however the same: a binary code that describes the local texture pattern is built by thresholding a neighborhood by the gray value of its center

Texture is an important descriptor in a CBIR system. Due to its ease of implémentation and prolific performance LBP based texture extraction is extensively used in the proposed system This extraction algorithm is performed over the subset of images selected from the first level of the retrieval process.

$$LBP_N = \sum_{i=0}^{N-1} f(P_i - CP)2^i; \quad f(p) = \begin{cases} 1; P \geq 0 \\ 0; P < 0 \end{cases}$$

Code snippets

```
query1=imread('querygray.jpg');
lbpFeatures = extractLBPFeatures(query1)

array2=[];
FileList2 = dir(fullfile('wangselectgray', '*.jpg'));
F2 = natsortfiles({FileList2.name});

for iFile = 1:numel(F2)
  File2 = fullfile('wangselectgray', F2(iFile));
  Img2  = imread(cell2mat(File2));
  lbpFeatures1 = extractLBPFeatures(Img2);
  A=abs(lbpFeatures-lbpFeatures1)
  B=sum(A);
  array2=[array2,B];
```

Vector answer

ans

1x883 single

	1	2	3	4	5	6	7	8	9	10
1	1.4418	2.0343	2.9845	2.1859	3.3051	3.0978	2.4987	1.7402	2.8793	2.1951
2										
3										
4										
5										

Edge descriptor

For edge description we use Canny edge detection method. The method works as follows. From the wang dataset the images (RGB) are converted to HSV as Canny edge detection Is applied on the V channel of the HSV. After which the HSV is combined back and converted to the RGB. Thus, here we obtain the edges RGB which can be operated on to identify the edges similarity.

In RGB color space, each color channel is highly correlated with other color channels such that the splitting of chrominance and luminance information is impossible and is perceptually ambiguous to human perception. Gray scale information is enough to mark edges in images but gray to RGB conversion is not possible to produce color image.

To overcome this color space transformation is performed to obtain the edge details from the intensity plane of an image. This is the first step in edge feature extraction process.

H and S carry the chrominance details of the given image. V channel holds the intensity distribution of that image. Canny edge detection algorithm is run over the V channel. Then, the edge extracted V channel is combined with un-modified H and S channel and transformed back to RGB color space

Code snippets

```
function cannyedge()
FileList3 = dir(fullfile('wangselect', '*.jpg'));
F3 = natsortfiles({FileList3.name});
for iFile = 1:numel(F3)
  File3 = fullfile('wangselect', F3(iFile));
  Img3  = imread(cell2mat(File3));
  [H S V]=rgb2hsv(Img3);
  a=edge(V,'Canny');
  b=hsv2rgb(H,S,a);
  filename3=sprintf('wangselectedge/myimage%02d.jpg',iFile);
  imwrite(b,filename3);
```

Main image

```
query2=imread('queryedge.jpg');
Red = query2(:,:,1);
Green = query2(:,:,2);
Blue = query2(:,:,3);
```

Query image

```
File3 = fullfile('wangselectedge', F3(iFile));
Img3  = imread(cell2mat(File3));
Red1 = Img3(:,:,1);
Green1 = Img3(:,:,2);
Blue1 = Img3(:,:,3);
```

Comparing R, G and B

```
C=abs(Red-Red1);
D=Red+Red1;
E=C.^2 ./D;
F=E./2
fmat=reshape(F,1,256*256);
```

```
G=abs(Green-Green1);
H=Green+Green1;
I=G.^2 ./H;
J=I./2
jmat=reshape(J,1,256*256);
```

```
K=abs(Blue-Blue1);
L=Blue+Blue1;
M=L.^2 ./L;
N=M./2
nmat=reshape(N,1,256*256);
```

And finally we save the sum of the three into one array as their sum.

It gives RGB edge images as the output as the following.

Similarity measure

We get two feature arrays one from the LBP and the second one from the comparing the edges extracted from the Canny edge detection. After which we combine them both by taking their average.

Before that we normalize them, by following.

```
normarray2=[];
normarray2=(array2 - min(array2))./(max(array2)-min(array2))

normarray3=[];
normarray3=(array3 - min(array3))./(max(array3)-min(array3))
```

```
array4=[];
array4 = normarray2./2 + normarray3./2

array5=[];
array5=sort(array4)
```

Similarity measure is mandatory in all kinds of the retrieval system as it gives the measure of distance between the low- level visual contents of two images. Distance information is the deciding factor of the similarity measure. The very low resultant value indicates that corresponding database image is very close to the given query image. LBP and edge features similarity are estimated through the Chi square distance measure which is given by the two equations below.

$$LBP\_SM(QLBP_{IMAGE}, NewDBLBP_{COUNTIMAGES}) = \sum_{i=1}^{N} |f_{QLBP_{IMAGE}}(i) - f_{NewDBLBP_{COUNTIMAGES}}(i)|$$

and

$$\begin{aligned} EDGE\_SM(QEDGE_{IMAGE}, NewDBEDGE_{COUNTIMAGES}) = & \sum_{i=1}^{N} |f_{QEDGE\_R_{IMAGE}}(i) - f_{NewDBEDGE\_R_{COUNTIMAGES}}(i)| \\ & + \sum_{i=1}^{N} |f_{QEDGE\_G_{IMAGE}}(i) - f_{NewDBEDGE\_G_{COUNTIMAGES}}(i)| \\ & + \sum_{i=1}^{N} |f_{QEDGE\_B_{IMAGE}}(i) - f_{NewDBEDGE\_B_{COUNTIMAGES}}(i)| \end{aligned}$$

These is also shown in code snippet form in the image no.

In the estimated normalized feature values are combined using equal weights and are organized in ascending order using bubble sort to achieve homogeneous images at the top level of the retrieval. Furthermore, the proposed CBIR system is probed with the standardized performance measures over the different number of retrieval results.

Algorithm

CTEBIR (Q_{IMAGE}, $DB_{NIMAGES}$) – Color, Texture and Edge Based Image Retrieval.

Input: Query image from the user is denoted by $Q_{IMAGE.}$ Image Database (DB) contains N number of images denoted by $DB_{NIMAGES}$. N denotes the total number of images.

Output:

K **number of images** $IM_1, IM_2, \ldots, IM_K \in DB_{NIMAGES}$ **retrieved as similar images for the given query image** Q_{IMAGE}

1. Separate the R, G and B color channel information from Q_{IMAGE} and $DB_{NIMAGES}$

2. Calculate the 1st and 2nd order color moments for the Q_{IMAGE} and form low and high threshold using Eqs. (3) and (4)

3. Calculate the 1st order color moment for images $\in DB_{NIMAGES}$

4. Count=0;

5. for i=1 to N do

a. if the 1st order CM of $DB_{iIMAGES}$ lies between the low and high threshold of the Q_{IMAGE} then select the particular image (IM) and store that in New Database (NewDB)

b. Count=Count+1;

C. NewDB$_{COUNTIMAGES}$ =IM;

d. end if end

6. Calculate LBP texture feature for Q_{IMAGE} and $NewDB_{COUNTIMAGES}$

7. Calculate Canny edge feature for Q_{IMAGE} and $NewDB_{COUNTIMAGES}$

8. for each image in $NewDB_{COUNTIMAGES}$ do

a. Measure the similarity metric between Q_{IMAGE} and $NewDB_{COUNTIMAGES}$ using normalized texture and edge feature

b. Combine the calculated similarity metric of the texture and edge feature of the images $\in$ $NewDB_{COUNTIMAGES}$ end

9. Bubble sorting is performed on $NewDB_{COUNTIMAGES}$ according to the value of each image which is obtained from step 8(b)

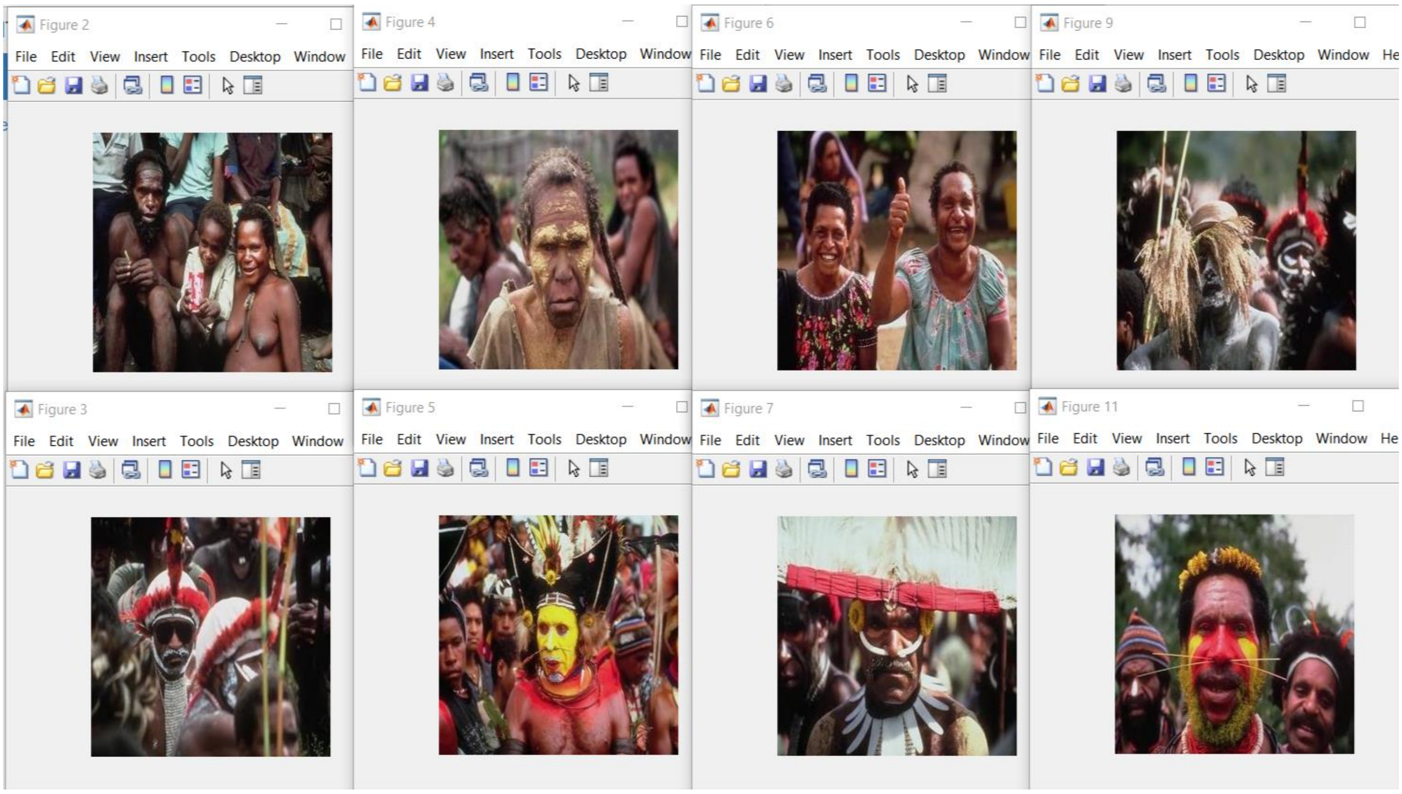
10. return K number of similar images $IM_1, IM_2, ..., IM_K$

Result:

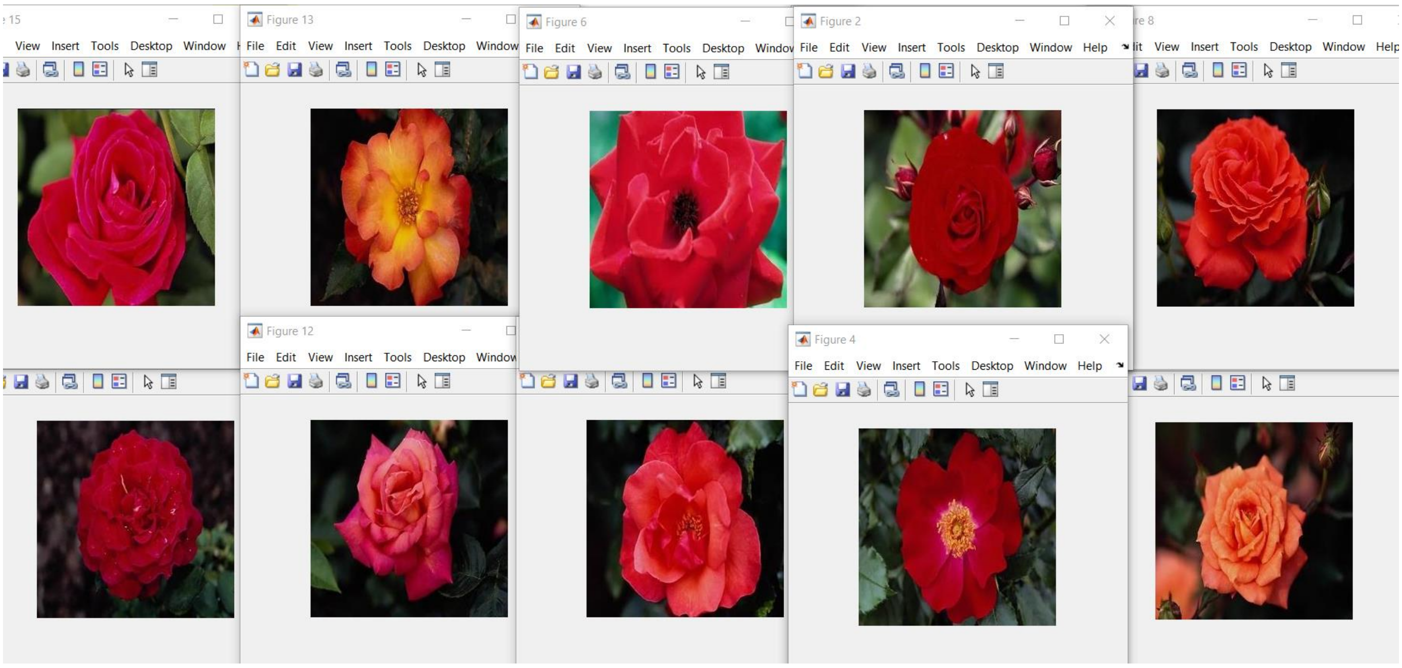
```
>> projectmain
What dataset u want to choose? - (enter corel10k OR corel5k OR wang) wang
```

Query image 1: -

Result 1: -



Query image 2: -



References

[1] L.K. Pavithra, T. Sree Sharmila .An efficient framework for image retrieval using color, texture and edge features

[2] Yue J, Li Z, Liu L, Fu Z. Content-based image retrieval using color and texture fused features. Math Comput Model 2011;54:1121–7. doi:10.1016/j.mcm. 2010.11.044.

[3] Lin CH, Chen RT, Chan YK. A smart content-based image retrieval system based on color and texture feature. Image Vis Comput 2009;27:658–65. doi:10.1016/j.imavis.2008.07.004.

[4] Kiranyaz S, Ferreira M, Gabbouj M. Automatic object extraction over multiscale edge field for multimedia retrieval. IEEE Trans Image Process 2006;15(12):3759–72. doi:10.1109/TIP.2006.881966.

[5] Agarwal S, Verma AK, Dixit N. Content based image retrieval using color edge detection and discrete wavelet transform. In: IEEE Int Conf on Issues and Chall in Intell Comput Tech (ICICT); 2014. p. 368–72. doi:10.1109/ICICICT.2014.6781310.

[6] Liu GH, Yang JY. Content-based image retrieval using color difference histogram. Pattern Recognit 2013;4(1):188–98. http://dx.doi.org/10.1016/j.patcog. 2012.06.001.

[7] Dey M, Raman B, Verma M. A novel colour- and texture-based image retrieval technique using multi-resolution local extrema peak valley pattern and RGB colour histogram. Pattern Anal Appl 2015. doi:10.1007/s10044-015-0522-y.

[8] Srivastava P, Binh NT, Khare A. Content-based image retrieval using moments of local ternary pattern. Mob Netw Appl 2014;19(5):618–25. doi:10.1007/ s11036- 014- 0526- 7.

. [9] Irtaza A, Arfan Jaffar M, Aleisa E, Choi TS. Embedding neural networks for semantic association in content based image retrieval. Multimedia Tools Appl 2014;72:1911–31. doi:10.1007/s11042-013-1489-6.

. [10] Wang XY, Yu YJ, Yang HY. An effective image retrieval scheme using color, texture and shape feature. Comput Stand Interfaces 2011;33:59–68. https: //doi.org/10.1016/j.csi.2010.03.004.