Image Search using Overlapping of Different Image Features

Vinod Kumar
Dept of computer Engineering
Delhi Technological University
(Formerly Delhi College of Engg.)
Delhi, India

Asheesh Kumar
Dept of computer Engineering
Delhi Technological University
(Formerly Delhi College of Engg.)
Delhi, India

Abhishek Bhardwaj
Dept of computer Engineering
Delhi Technological University
(Formerly Delhi College of Engg.)
Delhi, India

ABSTRACT

In this paper, a efficient method of 'Image Search', is proposed to find the most similar images for the given query image from the image Repository in which we first pre-process the original image based on different image features like edge, saturation, hue, brightness, luminance and then we will perform some similarity test on each features set of images and then find image according to the given algorithm of this paper that is find image which overlapping maximally at particular priority value, by taking all sets together, than display result.

Keywords

Image search, image factor, similarity test, overlapping, priority**.**

1. INTRODUCTION

Valuable information can be hidden in images. Plenty of knowledge can be extracted from these images data, So many people are devoted themselves into image retrieval research to improve current image search technique. The need for image mining is high in view of the fast growing amounts of image data [1].

Due to the digitization of image data and modernization in computer technology, it has become very easy to obtain, process and store large amount of data, particularly Multimedia data. Presently, tools for images search are few and require human intervention. Feature selection and extraction is the pre-processing step of Image Mining. Obviously this is a critical step in the entire scenario of Image Mining Process [2].

Image mining is the interdisciplinary which require effort from other computer field like image mining, computer vision, digital image processing, data mining and artificial intelligence.

Due to the advance in computer technology, now the information is communicated in form of the images. Image database is very large in size but now we have advance digital storage device which can store large amount of data efficiently. So when we work on images we focus on the process on image than storing of image.

In this paper we are using algorithms to retrieve image from repository using different image features. Image retrieval using multiple image features like saturation, brightness, edge etc is an efficient method. Image search based on image property becomes more desirable for developing large volume image retrieval applications

1.1 Different Image features

Single image consist lot features which can be used as information for the image retrieval system. Now we are considering following image features [3].

1.1.1 Hue

It is predicated on the principle that every real color originates from a single pure color (Hue), which is then mixed with various amount of white or/and black color to give various shades of that pure color. Hue is the name or pure value of the color such as red, green, yellow, etc. It is measured in degrees from 0 to 360 (0 is Red, 60 is Yellow, 120 is Green, 180 is Cyan, 240 is Blue and 300 is Magenta.).

1.1.2 Saturation:

Saturation is the purity of the color and is the amount of pure color mixed with white color. It varies from white to pure color. It is measured in percent from 0 to 100. The higher the percentage, the more pure will be the color.

1.1.3 Brightness:

Brightness determines the intensity of the color and is the amount of pure color mixed with black color. It varies from black to pure color. It is measured in percent from 0 to 100. The higher the percentage, the brighter the color. it is calculated in term of intensity.

1.1.4 Luminance (Y)

The amount of energy perceive by an observer, measured in candela per square metre (cd/m^2). Often the term *luminance* is used for the relative luminance, Y/Y_n, where Y_n is the luminance of the reference white point.
luminance can be calculated from linear RGB components:

$$Y = 0.2126 R + 0.7152 G + 0.0722 B$$

1.1.5 Edge of image

Edge information of a colour image is an important visual information to the detection and recognition of objects in an image. Edge of an image is sharp change in pixels value from nearby pixels value in image. Edges can be grouped to form regions.

1.2 Euclidean Distance

In Euclidean distance based on color image segmentation technique, the RGB color model is considered. In the RGB color model, each color appears in its primary spectral components of red, green and blue. Each RGB color pixel is a triplet of values namely Red, Green and Blue. Segmentation provides better results in RGB color model when compared to other color models.

Segmentation in color domain is based on similarity detection rather than discontinuity based. Similarity based detection directly groups the similar pixels [7].

The algorithm involves the following steps.

1) The first step involves take image frames.

2) On each frame the following operations are performed.

a) Take an average of R, G, and B color value of each segmented.

b) Euclidean distance measure between segmented values of images. The Euclidean distance between the image pixel 'I' and 'J' is

$$D\,(I, J) = \sqrt{[(IR - JR)^2 + (IG - JG)^2 + (IB - JB)^2\,]}$$

Any image pixel 'I' of an image is said to be similar to image pixel 'J' of another image if the Euclidean distance between them is less than a specified threshold D_0. Choosing D_0 is dependent on the defect that is to be classified. For all the thermographs of same defect, the value of D_0 is same, hence making this algorithm image independent and parameter independent.

1.3 Comparison of other retrieval technique

Most conventional image databases are text based. As a result, image retrieval from repository is based on text keyword searching. Text keywords of images are simple and very easy to manipulate. There are two major problems with this method.

1. Creating keywords for large number of images is time consuming.

2. Keywords are inherently subjective and not unique.

Other most important image retrieval technique is Content Based Image Retrieval (CBIR), which based on the visual content of the image. Image feature extracted based on the visual content of the digital image. In the CBIR color, texture, pattern, shape of object and location are the visual content of the image. CBIR give better result if image search based on the content. It requires lot of effort of image processing, which make it complex with other image retrieval techniques.

Due to these disadvantages, Image retrieval using multiple image features like saturation, brightness, edge etc is an efficient method. Image search based on image property becomes more desirable for developing large volume image retrieval applications

2. PROPOSED METHOD

The management of a large number of image data in a multimedia database has received much attention in recent years. Most of the earlier works are largely focused on techniques to extract useful information (such as color, texture or shape) that represents images in useful way. Rapid retrieval is becoming important issue as image data continue to grow in size and a slow will no longer be acceptable to the user community.

So in this paper we proposed an algorithm, in which we will first pre-process the original set of images on certain factor like hue, saturation, Brightness, edges, Luminance, etc. See figure 1.

We can also consider segmented images, texture, and color value as factors but to keep our demonstration simple we will consider simple factors like edge value, saturation value, brightness, hue, and luminance.

For all factors, we will generate different image sets, corresponding to each factor taken.

After pre-processing we will then select the image to be compares and will apply similarity test within same image set.

Then we will note down the result as per degree of similarity value, for different result set. Most similar image will be given priority 1, and then next one will be of priority 2 and so on.

After finding priority values for each image set ,we will traverse priority wise and we will now check the maximally occurring image at given priority and will add that image to our result , as per rules proposed in algorithm.

Finally we will get the set which represent the set of images having similar to chosen images in order from most similar to least similar.

Note that our result is based on chosen factors. More appropriate will be chosen factors, better will be results.

2.1 Assumptions:

- Here we are assuming that images in repository are animated images.

2.2 Algorithm:

<Preprocessing Step>

Step 1: Generate various sets of images. Each set contains pre-processed form of original Image based on certain factors. These factors include hue, saturation, brightness, luminance, Edge.

$IMAGE_i$ = <Different set of images pre-processed on certain factor>,

Where i € Total number of factors considered.

<Similarity Test over each $IMAGE_i$ set>

Step 2: Now select the corresponding pre-processed form of the image from each $IMAGE_i$ set, which you want to compare and perform similarity test over each $IMAGE_i$ set.

Step 3: Note the prioritization order of images, which is result of each similarity test, and save it in $RESULT_i$ set in ascending value of priority value. Images with highest Similarity value have priority 1, and then next image will have priority 2 and so on. i.e.

$RESULT_i$ =Similarity Test ($IMAGE_i$), where i € Total number of factors considered.

Note: For demo purpose, our similarity test will be as follows. First resize the image into 300*300 pixels. Then divide the whole image into 5*5 regions, and each region containing 30*30 pixels. To find similarity, we will then calculate the Euclidean distance between the regions and accumulate. Smaller will be Euclidean distance, higher will be similarity.

<Overlapping image at particular priority in different $RESULT_i$ set>

Step 4: Set TEMP_PRIORITY = 1

Set SIMILAR_IMAGE = <NULL>

Where TEMP_PRIORITY is a variable keeping track of current priority number and SIMILAR_IMAGE is a set of similar images as a result of our algorithm, i.e. set of images similar to chosen image.

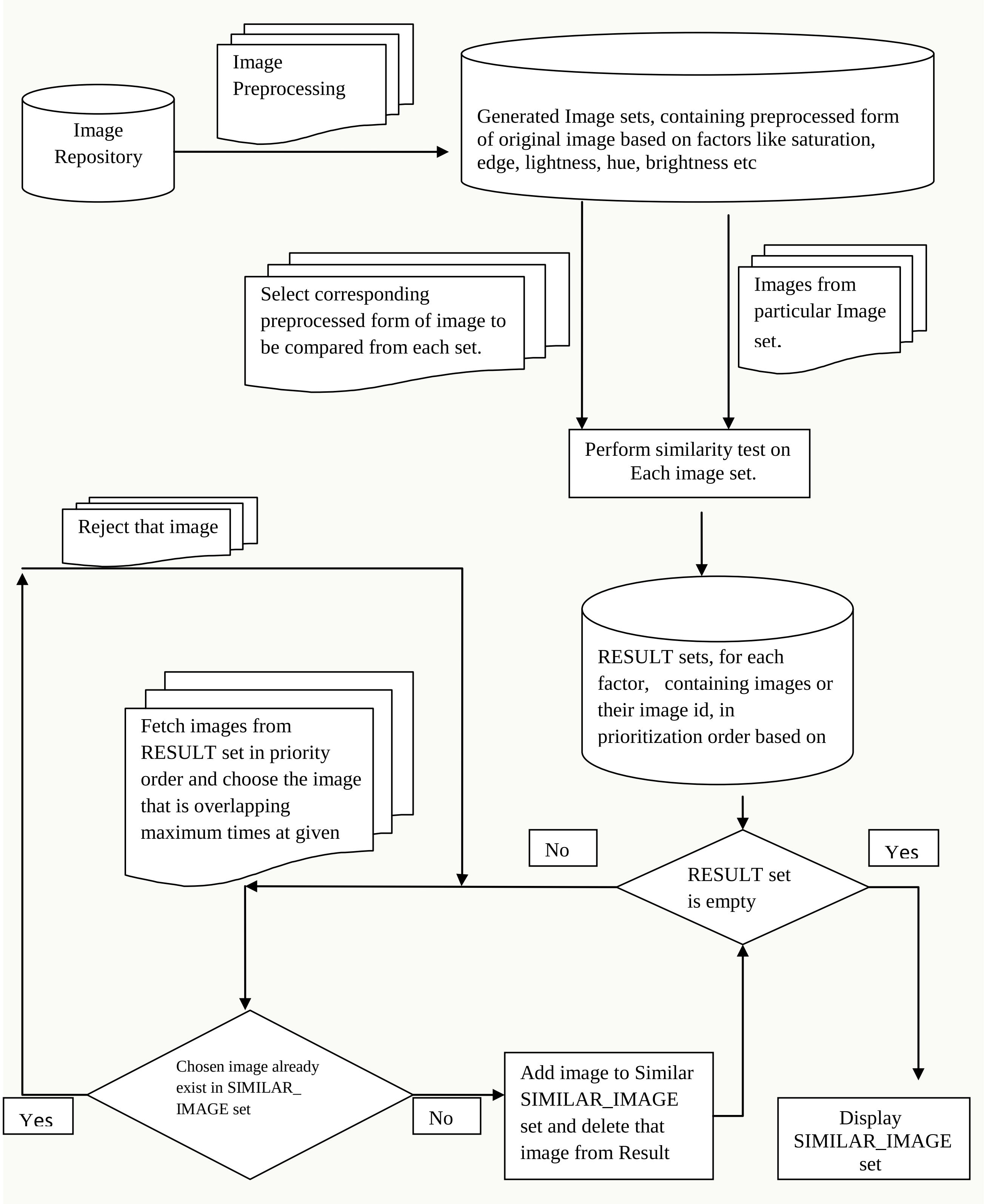
Image Repository
Image Preprocessing
Generated Image sets, containing preprocessed form of original image based on factors like saturation, edge, lightness, hue, brightness etc
Select corresponding preprocessed form of image to be compared from each set.
Images from particular Image set.
Perform similarity test on Each image set.
Reject that image
RESULT sets, for each factor, containing images or their image id, in prioritization order based on
Fetch images from RESULT set in priority order and choose the image that is overlapping maximum times at given
No
RESULT set is empty
Yes
Chosen image already exist in SIMILAR_ IMAGE set
Yes
No
Add image to Similar SIMILAR_IMAGE set and delete that image from Result
Display SIMILAR_IMAGE set

Fig1. Block diagram of searching image in repository based on overlapping method.

Step 5: While all RESULT$_i$ sets are not empty,
REPEAT steps (a) to (e)
a) Fetch the image$_j$ from all RESULT$_i$ sets that is overlapping maximum times at given priority, having priority = TEMP_PRIORITY. Here image$_i$ is a image variable from RESULT$_i$ set.
b) If image$_j$ already exist in SIMILAR_IMAGE set,
i) From all result set RESULT$_i$, delete image$_j$ i.e.
RESULT $_i$ = RESULT $_i$ – image$_i$,
ii) goto step (a).
c) If two image$_j$ eg. x, y have same no of occurrence then put it in waiting queue W. leave SIMILAR_IMAGE$_j$ output image blank for that RESULT$_i$ and set l=j.
i) Go for next row and step a.
ii) If maximum occurred image x is already in W queue, put y into SIMILAR_IMAGE$_l$ set and x in SIMILAR_IMAGE$_j$.
d) SIMILAR_SET = SIMILAR_SET + image$_j$.
e) TEMP_PRIORITY = TEMP_PRIORITY + 1.

STEP 6: Display SIMILAR_IMAGE set as final result.

Example

We are taking an example to clarify our approach. Initially we have set of Color animated images. Now we will pre-process each image on various factors like edge, saturation, intensity etc and save it in IMAGE $_i$ set correspondingly.

IMAGE$_i$ = <Different set of images pre-processed on certain factor>,
Where i is a total number of factors considered.

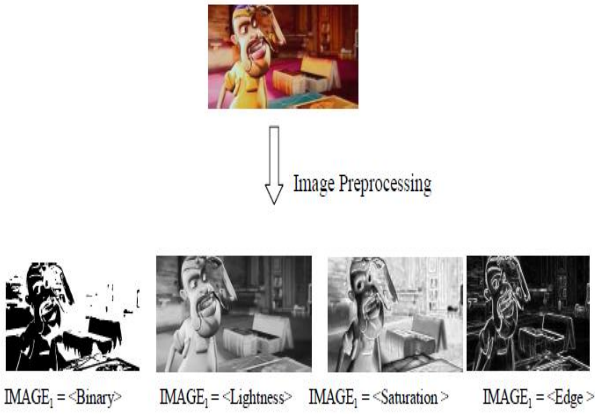

Fig 2: Diferrent image component of colour image

Now select the image that you want to compare and perform similarity test and save result in RESULT $_i$ set i.e.
RESULT$_i$ = Similarity Test (IMAGE$_i$),

- RESULT 1 based on Edge Similarity Test
  - Pic5
  - Pic17
  - Pic7
  - Pic18
  - ...
  - ...
- RESULT 2 based on luminance Similarity Test
  - Pic5
  - Pic7
  - Pic18
  - Pic17
  - ...
- RESULT 3 based on Saturation Similarity Test
  - Pic5
  - Pic7
  - Pic6
  - Pic18
  - ...
  - ...
- RESULT 4 based on Brightness Similarity Test
  - Pic5
  - Pic7
  - Pic18
  - Pic17

  …
- RESULT 5 based on Hue Similarity Test
  - Pic5
  - Pic7
  - Pic18
  - Pic19

  …

Now we will fetch images from result set RESULT$_i$, which is maximally overlapping at given priority value and will add it to similar image set SIMILAR_IMAGE$_{i.}$ as per algorithm.
Priority value and will add it to similar image set SIMILAR_IMAGE$_{i.}$ as per algorithm.

Edge	Saturation	Luminance	Brightness	Hue	
Pic5	Pic 5	Pic 5	Pic 5	Pic 5	R1
Pic 17	Pic 7	Pic 7	Pic 7	Pic 7	R2
Pic 7	Pic 6	Pic 18	Pic 18	Pic 18	
Pic 18	Pic 18	Pic 17	Pic 17	Pic 19	
Pic 8	Pic 8	Pic 6	Pic 6	Pic 9	
Pic 6	Pic 17	Pic 8	Pic 8	Pic 8	
Pic 25	Pic 21	Pic 25	Pic 10	Pic 25	
:	:	:	:	:	

Table 1:Result of similar image on different image

R1: Pic5 is maximally overlapping at priority 1, so add it to SIMILAR_IMAGE set.

R2: Pic7 is maximally overlapping at priority 2, so add it to SIMILAR_IMAGE set.

R3: Pic18 is maximally overlapping at priority 3, so add it to SIMILAR_IMAGE set.

R4: Pic17 and Pic18 have same number of occurrence at priority 4, but Pic18 already exists in SIMILAR_IMAGE set, so add Pic17 to SIMILAR_IMAGE set.

R5: Pic8 and Pic6 are maximally overlapping at priority 5, Using algorithms we check both occurrence at next priority row.

R6: Pic8 and Pic6 are maximally overlapping at priority 6, but Pic6 already exists in SIMILAR_IMAGE set, so we will delete it from RESULT set from this priority and will look for next maximally overlapping image at priority 6, which is Pic8, so add Pic8 to SIMILAR_IMAGE set.

Final Result

In the above table we will now evaluate each row and will find maximally occurring image at each row and will add it to SIMILAR_IMAGE set. If the image already exists in SIMILAR_IMAGE set then we will evaluate next maximally occurring image at that row.

SIMILAR_IMAGE set
Pic5
Pic7
Pic18
Pic17
Pic6
Pic8
Pic25

Table 2: Result output

3. CONCLUSION

The dramatic rise in the sizes of images databases, rise a need of the development of effective and efficient retrieval systems. The development of these systems started with retrieving images using color feature is not sufficient. Systems search images from repository using Overlapping approach in given visual features such as color, texture and shape, as opposed to depending on image descriptions or textual indexing. In this paper, we have researched various modes of representing and retrieving the image properties of color, and image feature. The application performs a simple search in an image dataset for an input query image, using image feature such as edge, saturation, luminance etc. It then compares the query image with data set image on different images feature using the Euclidean Distance Equation. It is shown that our method gives better results than taking color only.

4. FUTURE WORK

The work presented in this paper can be further enhanced by finding the appropriate factors which represents the image uniquely. If we search more appropriate features of image then there will be larger degree of similarity between matching images, hence more appropriate will be the results.

5. REFERENCES

[1] Hu Min and Yang Shuangyuan . Overview of Image Mining Research. In Proceedings of the 5th International Conference on Computer Science & Education Hefei, China. August 24–27, 2010

[2] Patricia G. Foschi, Deepak Kolippakkam, Huan Liu and Amit Mandvikar. Feature Extraction for Image Mining.

[3] R.C Gonzalez, R.E Woods. Digital Image Processing (2nd Edition).

[4] Ji Zhang, Wynne Hsu and Mong Li Lee. Image Mining: Issue, Framework and Technique.

[5] Peter Stanchev Kettering University Flint, Michigan. Using Image Mining for Image Retrieval. In Proceedings of the Conference on Computer Science and Technology, Maxico May 19-21 2003.

[6] Aura Conci, Everest Mathias and M. M. Castro .Image Mining by Color Content.

[7] Yu changjinand Xia hongxia "The Investigation of Image Mining Framework" at 2009 IEEE

[8] Dr V Mohan, A Kannan" Color Image classification and retrieval using image mining technique"

[9] Dengsheng Zhang and Guojun Lu. Evaluation of Similarity Measurement for Image Retrieval

[10] Chin Chen-Chaur, Ting Chu Hsueh. Similarity Measurement between Images, In Proceedings of the IEEE Annual International Computer Software and Applications Conference COMPSAC'05), pp.207-248, 2005.

[11] Timothy K.Shih,Jung-yao Huang,Ching-Sheng Wang,Jason C.Hung And Chuan-Ho Kao. Intelligent Content-Based Image Retrieval System Based on Color, Shape and Spatial Relations.

[12] Rajshree S.Dubey, Niket Bhargava, Rajnish Choubey, 2007.Image Mining using Content based Image retrieval System