International Journal of Advanced Science and Technology
Vol.58, (2013), pp.51-64
http://dx.doi.org/10.14257/ijast.2013.58.05

Image Retrieval System for Composite Images using Directional Chain Codes

Akriti Nigam*, Rupesh Yadav* and R. C. Tripathi*

**Indian Institute of Information Technology, Allahabad*
akriti.nigam@yahoo.co.in, 7yadavrupesh@gmail.com, rctripathi@iiita.ac.in

Abstract

Most of the image retrieval systems published and implemented have focused on basic features like color, shape and texture of an image with little or no consideration of the included text region of the image. In this work, we have developed an image retrieval system that works on finding similar composite images containing graphical shapes as well as text from a database of thousands of images. By proposing a novel method for text localization, extraction followed by detection, we have demonstrated how this method outperforms commercial OCR tools. The significant feature of this work is its handling the requirements of invariance to font size, design, text region orientation and its ability to give accurate result even in the presence of complex background and graphical elements. The methodology has been tested for English text but is capable to handle any other language.

Keywords: *Text extraction, Image retrieval, Directional chain codes, connected components*

1. Introduction

Text recognition is a field of work having vast applicability. Typical applications include simple text extraction of rare manuscripts for digital libraries to number plate text extraction for traffic surveillance, text detection for natural images as well as in videos having cluttered backdrops. In this paper, the focus is on the problem of extracting and detecting text from composite images. Composite images represent images that contain some graphical element along with text part either with the graphical design within the text part or outside it.

Search for information in this digital age is slowly shifting from text to images for *e.g.*, the Google image search now allows the user to simply upload the image of the object, person or place they want to search about instead of searching based on text entry comprising of keywords. But what is essential for getting effective results from image search is deciding how to extract the most powerful features of the images in consideration. While extracting features most published works focus on content based image retrieval (CBIR) considering the image as a whole [17, 18]. Even segmenting the image into different areas and extracting local features may not give good results if the image contains text also.

In this paper we propose a novel technique for building content based image retrieval (CBIR) system that is based on finding images containing similar textual content within an image. This system basically focuses on-

- Extracting text from images.
- Recognizing extracted text.
- Retrieving similar images

ISSN: 2005-4238 IJAST

Examples of composite images are natural scene images, image of buildings, advertisement boards, trademark images *etc*. Composite trademark images generally contain images along with text comprising of the brand name or/and the product's slogan. The problem of text extraction from the latter type of images is difficult as compared to text extraction from number plates or manuscripts because of the fact that text in such composite images need not be in any standard font size or font family. It is usually symbolic and artistic. In some cases the text is especially artistically designed using graphic designing software packages or may be hand written. Handling this vast variety of text present in natural composite images requires a robust technique that should not depend completely on standard font templates.

Using OCR tools for text extraction from scenic images containing random background, usually gives poor result because most of the commercial OCR tools are programmed to work on scanned text images with standard fonts which makes it easy to separate out text characters from the background. Commercial OCR's work well on scanned images of text documents but fail when images have more complex background in which text may be embedded in an unstructured fashion.

2. Related Work

The problem of automatic character recognition can be broadly divided into three stages: text localization, text extraction and text recognition. A lot of research work has been done for developing different techniques for extracting and recognizing text in images but most of these works concentrate on images that contain structured background where text localization is easy. Most of the approaches published like [13, 14] focus on one of the three stages mentioned above; only a few provide a method that performs both localization and extraction of text [1, 8].

The method proposed by Yi-Feng Pan *et. al.*, [2] uses Random field model to label different connected components as either a text or non text component. A text confidence map has been used to classify regions into text and non text based on the conditional probability.

Jung-Jin Lee *et. al.*, [3] have shown in their work how six classifying features are used to represent each image in two publically available databases of natural images: ICDAR 2003 and from Microsoft Research India. AdaBoost is used to construct a strong classifier that works on the six classifying features.

Kim *et. al.*, [4] segmented the images based on color regions. Shapes that do not seem to be part of text regions are eliminated using some predefined elimination criteria. Further, all the horizontal text lines in the images are extracted from the image. Text segments are then extracted from individual text lines using several features like width, height, aspect ratio, surface ratio etc. The above method has some restrictions when it comes to text alignment and orientation. Only horizontal text lines can be extracted using above algorithm. Curved text shapes will fail in getting detected. A similar work reported earlier makes use of heuristic rules to remove non textual parts from the image in [12].

There are several works that focus on grouping similar intensity regions that are homogeneous and have similar properties. The ease of this approach makes it a popular approach but the efficiency is low as this method is sensitive to noise and cannot work well for artistically created text regions not from any standard font family. Some of the previous methods based on the concept of reducing the search space using a sliding window have been discussed in [3-5, 15].

A similar work based on sliding window has been discussed in [11]. The recognition pipeline starts with character detection and recognition followed by grouping them into words and finally the words are re- scored based on their overall layout to get meaning out of the words.

Microsoft Corporation had suggested novel approach [6] using the width of the strokes made to write each individual character for detection of text. Their work used this stroke width feature value to separate text region from a non text region. The result obtained is better than many previously reported algorithms.

Use of stroke width transform is also reported in [9] which employ an image operator function to determine the stroke width of connected components. For text detection in [9], maximally stable extremal regions have been employed to signify individual characters in the image.

In [7], Chucai Yi *et al.*, have proposed an algorithm for detection and classification of textual information in natural scene images. The method proposed in their work, extracts features using interest point detection of characters.

Some of the major recognition problems faced by most of the previous works [10] in this field include the difficulties of poor contrast of individual characters in the image, textual characters having a cluttered background, individual characters joined together or if the image contains only a single letter.

Wavelet transform based text detection is discussed in [13]. The work proposed in [13] is limited to detection of text aligned horizontally. The extracted regions are further refined to be submitted as input to a recognition system.

3. Proposed Work

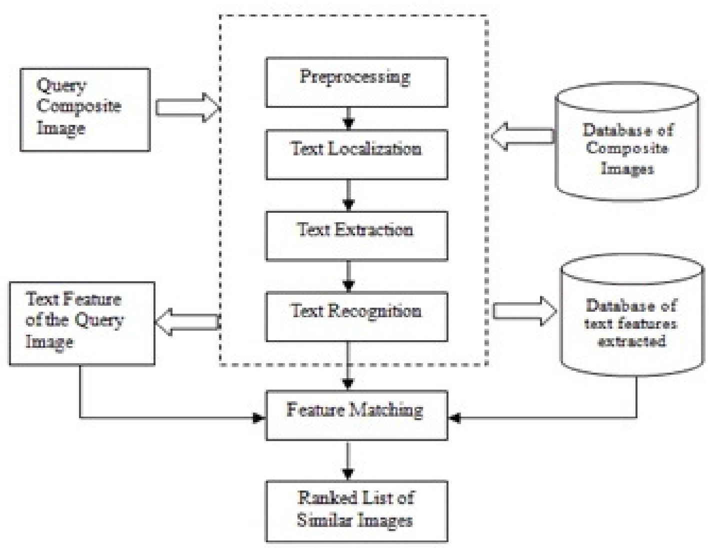


Figure 1. Flowchart of the Proposed System

3.1. Character Extraction

i. Start processing with the top most left black pixel point.

ii. Using this as the start pixel point we extract the entire region using connected components algorithm. Crop this region and save it as a separate image.

iii. After removing the component extracted in step ii, make that region as the background.

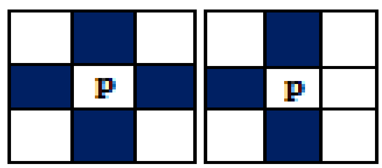
iv. Repeat steps 'i' through 'iii' still we have extracted and saved all disjoint regions in the image.

v. At the end of this function all the extracted components are presented as separate image files so that further processing steps can also be applied separately on each of them.

3.2. Boundary Extraction

For finding out the 8 directional codes for each character component, we first find out the boundary of the extracted component so that we can capture the essence of the boundary of the object. For this we apply first the edge extraction function on each component that we have extracted in Section 3.1. The idea is to obtain one pixel wide boundary of each component so that while running the direction code function, each point being processed has only one unvisited neighbour.

Following process is used to extract the boundary of all the character components. For each pixel point 'p' we check the intensity value of its 4-neighbours. If any one of the 4-neighbours of 'p' as shown by the shaded pixels in Figure 2 below, has value one (white pixel corresponding to background) then we will consider point 'p' as a part of the boundary otherwise we will ignore 'p' and move to the next pixel.



(a) (b)

Figure 2. a) Point 'p' is a Pixel Inside the Boundary b) Point 'p' is a Boundary Pixel

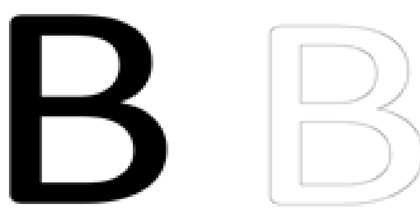


Figure 3. Original Character Component and its One Pixel Wide Boundary

3.3. Finding Direction Sequence

Now for each component extracted in Section 3.1, using its boundary image we shall find the direction sequence.

i. We start from the left upper most portion of every boundary image.

ii. Find its neighboring pixel point by moving in clock wise direction.

iii. When moving from one pixel point to its neighboring pixel point, store in an array the direction code [16] followed (using 8 standard directional chain code (DCC) shown in Figure 4).

iv. Repeat steps ‘ii’ to ‘iii’ until we return to the initial point from where we started.

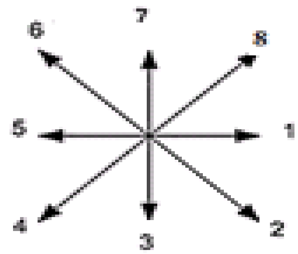


Figure 4. Standard 8 Directional Chain Code

3.4. Compressing the Direction Sequence

Directional chain codes have a disadvantage that they are too long for large size objects. For *e.g.*, consider the object in Figure 5 and the corresponding directional chain code. As we go on increasing the size of objects the chain codes will tend to become longer. To handle this problem, following method is used-

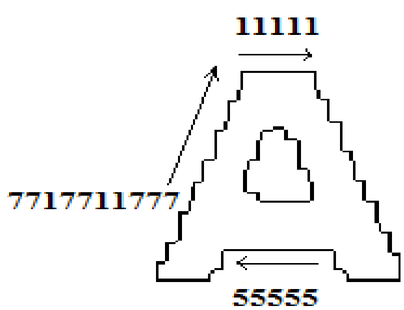


Figure 5. Directional Chain Codes along the Straight and Diagonal Edges of the Character

Each digit in the directional code is extracted from the sequence and sent to a function F that checks whether the direction code corresponds to a straight line or to a diagonal line. A threshold ‘T’ is decided beforehand. If we get ‘T’ or more than ‘T’ similar direction codes continuously, then the function outputs single digit of the same direction code.

*Diagonal edges do not have a single direction code instead they are a combination of two or three direction sequences as shown in Figure 5. Function F also checks for diagonal edges using this concept. If for e.g. we get a sequence of ‘35’ equal to or more than ‘T’ number of times then the function outputs ‘35’ once to represent a diagonal edge.

*Using this method the process becomes invariant to scaling. Two similar characters having different sizes will have same directional chain code after applying the direction code compression step mentioned in this section.

3.5. Rotation Invariance

To get accurate result of retrieval, the system should be invariant to any form of transformations like translation, scaling, rotation *etc*. The proposed approach is invariant to scaling which has already been shown in Section 3.4. Since chain codes are not affected by change in location of the object hence the invariance to translation is already obvious. To handle the third transformation i.e. making the retrieval invariant to rotation of the object, we modify the directional chain codes obtained in above section to compute the first difference and then the shape number from the chain code sequence. First difference is obtained by finding out the difference in direction between two consecutive directions in the sequence, in addition to this we find out the minimum integer value that can be obtained by considering the first difference sequence to be a circular chain. The sequence thus obtained is the shape number.

3.6. Standard 8-directional Chain Sequence of all English Alphabets

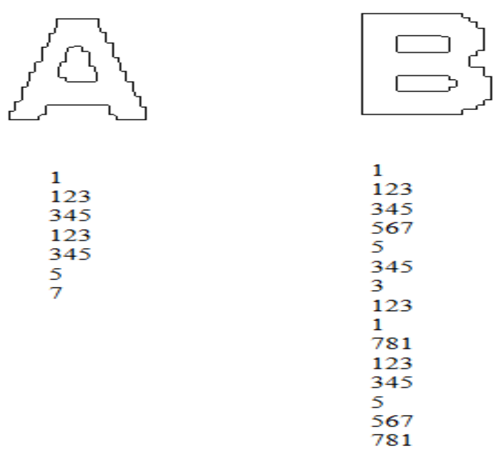


Figure 6. Standard DCC’s of some English Alphabets, Saved in the Text Feature Database

The database of text features contains the Directional chain codes of all the upper case and lower case English alphabets and 0-9 digits. Figure 6 above shows an example of the directional chain codes of letter A and B. Using these as the standard; we match the directional chain codes of different components in the query image and decide if a character is recognized or not.

3.7. Matching of the Directional Chain Code (DCC) Sequence

The matching function takes the directional chain code of each extracted component as an input and matches it with the standard DCC’s of all the alphabets and digits. For each match, the function builds a character match matrix and output's a percentage match value. Applying a threshold on the percentage match value, the system decides whether a character is recognized or not. This process is applied to all the extracted components and based on the characters recognized; the final output gives a doc file containing the textual string recognized in the query image.

The matching algorithm is liberal to one step change in directional chain code sequence matching *i.e.*, if at a particular step the directional chain code of input and database sequence do not match then the algorithm compares the same database chain code with next code of the input sequence. This way it ignores minor changes in boundary shape that may occur due to some noise.

The matching algorithm has been described in the table below-

Algorithm

Matching input direction sequence with all direction sequences in database (A-Z, a-z, 0-9)

I-Array [] - *Direction sequence of query image component.*
D-Array [] - *Standard direction sequence of characters stored in database.*
count- *number of matches between two direction sequences.*

```
for each character sequence stored in database
                    for every single direction code 'i' in D_Array[M]
                    for every single direction code 'j' in I_Array[N]
                              if  i==j
                                  i++;
                                  count++;
                                  miss=0;
                              endif
                              else if miss<1
                                  miss++,first_miss=j;
                              endif
                              else
                                  miss=0,j=first_miss,break;
                              endelse
                    endfor
          endfor
          // Calculate match_percentage //
          Match_Percent = (float)((count)*100)/((float)(N+M)/2));
          // Check with all previously stored Match_Percent //
          G_Check[Chr]=Match_Percent;
Endfor
//Find the closest matchinf character corresponding to maximum values in G_Check[] //
Closest_char=max (G_Check[Chr]);
```

Index	Direction sequence of query component	Direction sequence of characters in database	
0	1	1	Miss=0
1	123	123	Miss=0
2	567	5	Miss=1
3	5	567	
4	345	5	
5	781	345	
6	-	5	
7	-	781	

Figure 7. Progress of Matching Algorithm

3.8. Similar Image Retrieval

The system performs following steps for each query image-

i. Each query image is first preprocessed to remove noise using median filter and enhance edges in the image using erosion and dilation operations.

ii. The text feature of the image is extracted using the proposed method mentioned in Section 3.

iii. The string obtained as an output of the string matching function is compared with the text strings corresponding to all the database images.

iv. A similarity value is computed for each database image using-

$$R=\frac{\alpha}{\beta} \quad (1)$$

where α is the number of characters matched and β is numbers of characters that do not match.

v. Images corresponding to 10 highest values are retrieved in the final step.

4. Result

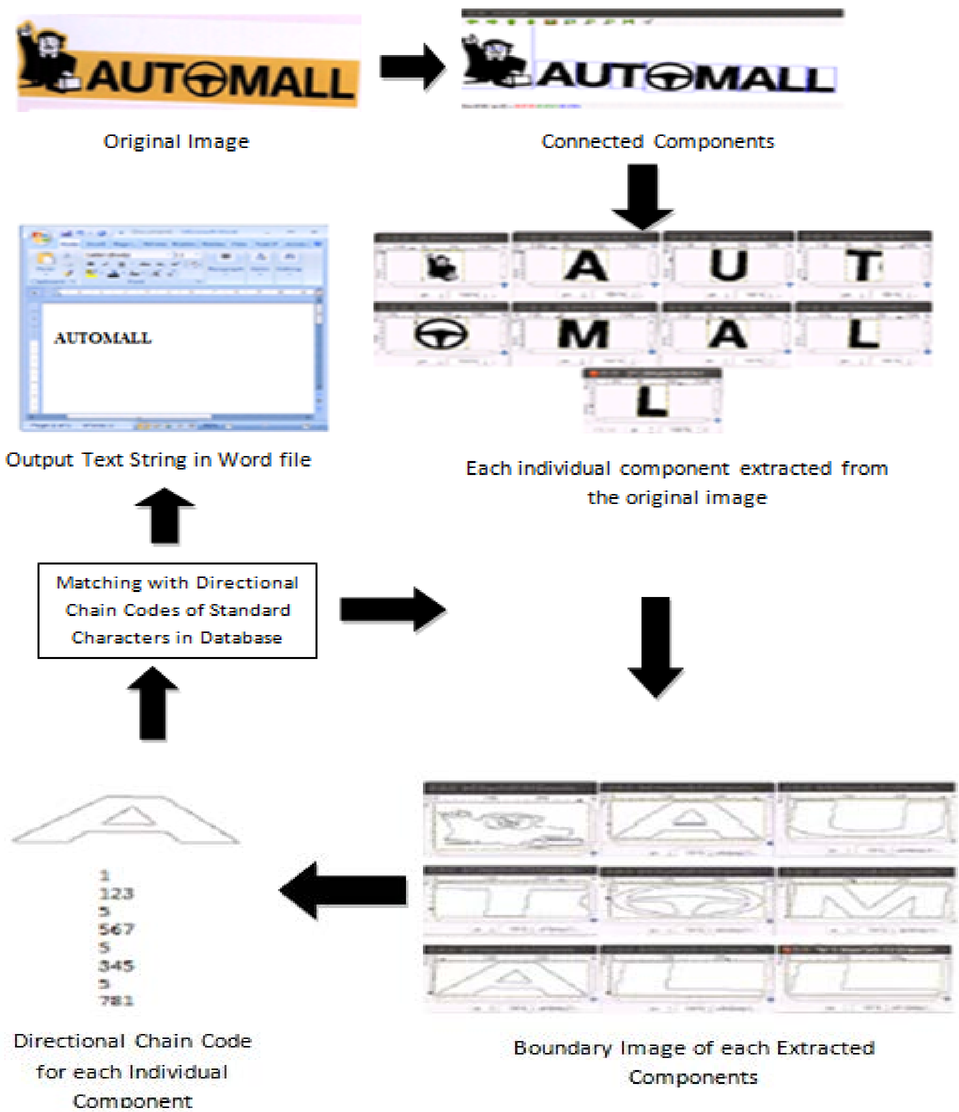


Figure 8. Flowchart Showing Results obtained at each Stage of Processing using Proposed Method

The Figure 8 above shows the result obtained at each step of processing discussed in section III. The result of character extraction on different images has been shown in Figure 9 for different alignments of text. We further compare the results of the proposed method of text extraction and recognition with the results obtained using some of the commercial OCR tools. The results of these OCR tools on different composite images are shown in Table 1. We can see the inefficiency of these tools when applied to some complex composite images.

Figure9. Original Image and the Corresponding Components Extracted. Third Image Shows Extraction of Text Components having Complex Alignment

Table 1. Results Obtained using Three Commercial OCR Tools compared to the Proposed DCC based System

IMAGES	Free-OCR	Abbyy Fine Reader 11 Professional Edition	New OCR	Free online OCR	Proposed DCC based system
	Q redfiat h M \	Not recognized	redhag ‘ T	No text found	redhat
	Not recognized	Not recognized	Not recognized	Not recognized	GoL

	%% JAVA	% ^	Ŵ A JAVA	% JAVÄ	JAVA
	Ő A% MASTERS		AV{i:£‘l ls” A		AVANTHA MASTERS

4.1. Recall and Precision

The table below shows the average recall and precision values obtained for different query images, where Precision and Recall are represented as following-

$$Precision = \frac{No.of\ relevant\ images\ retrieved}{Total\ No.of\ images\ retrieved} \quad (2)$$

$$Recall = \frac{No.of\ relevant\ images\ retrieved}{Total\ No.of\ relevant\ images\ in\ the\ database} \quad (3)$$

Table 1. Recall Precision Values for the Proposed System

	Recall	**Precision**
Query 1	95%	64%
Query 2	100%	72%
Query 3	100%	80%
Query 4	92%	64%
Query 5	98%	84%
Average	97%	72.8%

5. Conclusion

In this paper we have proposed a novel algorithm for text localization, extraction and recognition in composite images that contain major part as text along with some graphical elements. The method proposed is invariant to three basic forms of transformation i.e. translation, scaling and rotation. Previously reported image retrieval systems mostly focus on either color, texture or shape feature of the graphical object in an image. These features however are not appropriate for recognizing and using textual information in the image as a feature for matching with other images in the database. Hence we suggested this approach based on connected components and directional chain codes of textual regions.

The problems usually faced by OCR tools have been overcome in this proposed approach *i.e.*, the proposed approach works efficiently even in the presence of complex background. It gives accurate result even if the alignment of text is neither horizontal nor vertical. Similarly flexibility to font size, font family has been ensured along with ease of adding standard directional chain codes of characters in some languages other than English. The results show that the proposed text recognition system outperforms most of the commercially available OCR tools and the retrieval system achieves high precision and recall values.

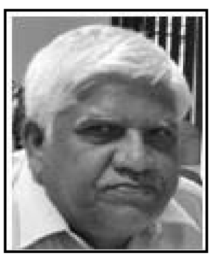
References

[1] C. Yi and Y. Tian, "Text extraction from scene images by character appearance and structure modeling", Computer Vision and Image Understanding, vol. 117, **(2013)**, pp. 182-194.
[2] Y.-F. Pan, X. Hou and C.-L. Liu, "Text Localization in Natural Scene Images based on Conditional Random Field", 10th International Conference on Document Analysis and Recognition, **(2009)**, pp. 6-10.
[3] J.-J. Lee, P.-H. Lee, S.-W. Lee, A. Yuille and C. Koch, "AdaBoost for Text Detection in Natural Scene", International Conference on Document Analysis and Recognition, **(2011)**, pp. 429-434.
[4] X. Chen and A. L. Yuille, "Detecting and reading text in natural scenes", CVPR, vol. 2, **(2004)**, pp. 366-373.
[5] R. Lienhart and A. Wernicke, "Localizing and segmenting text in images and videos", Circuits and Systems for Video Technology, vol. 12, no. 4, **(2004)**, pp. 256-268.
[6] B. Epshtein, E. Ofek and Y. Wexler, "Detecting Text in Natural Scenes with Stroke Width Transform", CVPR'10, **(2010)**.
[7] C. Yi and Y. Tian, "Text extraction from scene images by character appearance and structure modeling", Computer Vision and Image Understanding, vol. 117, **(2013)**, pp. 182-194.
[8] L. Neumann and J. Matas, "Text Localization in Real-world Images using Efficiently Pruned Exhaustive Search", IEEE International Conference on Document Analysis and Recognition, 1520-5363/11, **(2011)**, pp. 687- 691.
[9] H. Chen, S. S. Tsai, G. Schroth, D. M. Chen, R. Grzeszczuk and B. Girod, "Robust text detection in natural images with edge-enhanced maximally stable extremal regions", 18th IEEE International Conference on Image Processing (ICIP), **(2011)**, pp. 2609-2612.
[10] L. Neumann and J. Matas, "Real-Time Scene Text Localization and Recognition", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), **(2012)**, pp. 3538-3545.
[11] K. Wang, B. Babenko and S. Belongie, "End-to-End Scene Text Recognition", IEEE, International Conference on Computer Vision (ICCV), **(2011)**, pp. 1457-1464.
[12] A. Ikica and P. Peer, "An improved edge profile based method for text detection in images of natural scenes", IEEE EUROCON-International Conference on Computer as a Tool (EUROCON), **(2011)**, pp. 1-4.
[13] J. Gllavata, R. Ewerth and B. Freisleben, "Text Detection in Images Based on Unsupervised Classification of High-Frequency Wavelet Coefficients", ICPR, Proceedings of the 17th International Conference on Pattern Recognition, vol. 1, **(2004)**, pp. 425-428.
[14] J. Gllavata, R. Ewerth and B. Freisleben, "A Robust Algorithm for Text Detection in Images", IEEE, ISPA 2003, Proceedings of the 3rd International Symposium on Image and Signal Processing and Analysis, vol. 2, **(2003)**, pp. 611-616.
[15] M. Zhao and S. Li, "Sparse Representation Classification for Image Text Detection", IEEE, ISCID '09. Second International Symposium on Computational Intelligence and Design, vol. 1, **(2009)**, pp. 76-79.
[16] R. C. Gonzalez and R. E. Woods, "Digital Image Processing", Prentice Hall, **(2002)**.
[17] V. H. Kondekar, V. S. Kolkure and S. N. Kore "Image Retrieval Techniques based on Image Features: A State of Art approach for CBIR", (IJCSIS) International Journal of Computer Science and Information Security, vol. 7, no. 1, **(2010)**.
[18] S. Nandagopalan, B. S. Adiga and N. Deepak "A Universal Model for Content-Based Image Retrieval", World Academy of Science, Engineering and Technology, vol. 46, **(2008)**.

Authors

Akriti Nigam, received B. Tech degree in Information Technology from United College of Engineering and Research, Allahabad in 2005. In 2009 received M. Tech degree with specialization in Human Computer Interaction from Indian Institute of Information Technology, Allahabad and presently pursuing research from same university since 2011. Specialize in image processing, intellectual property rights, pattern recognition.

Rupesh Yadav, received his B.Tech. degree from IETE, Delhi, in Information Technology. He is presently pursuing M.Tech,(HCI) from Indian Institute of Information Technology, Allahabad. Areas of interest are Image Processing, Human Computer Interation, Graphics and Computational Intelligence.



Prof. R.C. Tripathi, received M.Sc. degree (Physics with Electronics) in 1967 and Ph.D in Solid State Physics using Electronics Instrumentation in 1972 from University of Allahabad. He is Ex. **Sr. Director, R&D** in IT Group, DIT, Min of Comm'n & IT, (MCIT) Govt of India, New Delhi and is presently Dean (SA) and Head of Department (IPR) in Indian Institute of Information Technology, Allahabad. His area of interest includes Digital Image Processing, Digital and Data Communications, Intellectual Property Rights, Computer Networks Management and Information and Network Security.