Online Information Search from Tamil Document Images in World Wide Web

Abirami.S
Department of Information Science and Technology,
Anna University, Chennai

Murugappan.S
Department of Computer Science and Engineering,
Annamalai University

ABSTRACT
Information Retrieval (IR) from Tamil document images present in World Wide Web (WWW) has become a challenging problem today due to its rising popularity. Among the most valuable Web assets, categorizing web images and retrieval of information from the images on the Web is quite difficult. This paper proposes a simple and effective method to separate the document images from the available web image sources and to retrieve the information present in those web document images. This system works in two phases: In the first phase, it concentrates on Automatic Image categorization process over web images by employing a filtering technique to discriminate the document images from other images available in WWW. Filtering technique employed here captures the image information by intensity and frequency histograms to discriminate the web document images. As for information retrieval in the second phase, feature string generation technique has been used to generate feature strings for every word images by extracting its shape this generates a feature string for every word image by extracting its features relying on their statistical properties, such as lines, black and white disposition rates and outline features of characters, instead of recognizing the letters and assigning its ASCII value like OCR. This kind of information retrieval has been initiated over a list of web sites and experimental results are recorded.

Keywords
Web Search, Information retrieval, Web Image categorization, Document Images.

1. INTRODUCTION
With the fast evolvement of Internet technology, the Internet has played an important role in our daily life. Websites offer information and services to their users. With the rapid development of Web, gaining useful information from large amount of information is more and more difficult. As resources in the internet are very large, it brings irrelevant information to users. In this case, the requirement for information is also changing all the time. Therefore, the search engine technology is introduced. Owing to, Universal search engines cannot meet the need of users to get specific information from the images. Due to the huge size of the Web, it has become difficult for the users to find their required information from images. Even though lot of tools in search engines are available today, to manipulate textual information from web documents, access to Web Document Images is in its infant stage.

In order to access the information present in web document images a suitable mechanism is required to discriminate the textual images from other images available in the Internet. Since guided by this motivation, this paper attempts to retrieve the information from Tamil document images available in WWW in twofold: 1. Automatic Web Image Categorization process 2. Information Retrieval Process. Automatic Web Image Categorization process separates the non document images from the document images available in WWW and information retrieval process captures the information present in the categorized document images and fetches the documents based on the user query.

2. RELATED WORK
Considering the highly visual and graphical nature of the World Wide Web, the number of image search engines is very limited. In the last few years, several image search engines like Google Image Search, WebSEEK [25], WebSeer [27], have been developed. While earlier image search engines used text only, WebSEEK and WebSeer use content based information like color, texture, shape etc. for indexing and query. However, very few tools are currently available for searching for images and videos. This absence is particularly notable [22, 24]. Visual information is published both as embedded in Web documents and as stand-alone objects. The visual information takes the form of images, graphics, bitmaps, animations and videos [14].

Region Based image search Engine is used for image collection, segmentation into regions and region feature extraction from real Web sites and for test images. Basically, this is based on Region Extraction. The segmented regions correspond to semantic objects, allowing efficient indexing and retrieval [16]. But most of the users are more interested in the identification of objects and actions depicted by images than in the color, shape, and other visual properties that most Content Based retrieval systems provide [26]. Because object and action information is more easily obtained from captions, caption-based retrieval appears to be the only hope for broadly useful image retrieval [26].

In 2002 Neil C. Rowe, proposed a intelligent agent Web crawler[20] and caption filter, searches the Web to find image captions and the associated image objects. He mainly searches the clues words from the captions of Meta data, and other text clues except captions. It uses a broad set of criteria to yield higher recall than competing systems, which generally focus on high precision [20].

In 2008, An Image Categorizer was introduced that categorizes logo and Trade mark images in the Web Images. It uses the image content features to categorize the Web images [23]. But no search engines till now isolates the images as textual (document images) and non textual images. Therefore this paper attempts to separate the non textual image as a first process. As a Second phase, an attempt has been made to capture the information present in the document images available in WWW.

Accessing collections of document text is a problem that has been addressed by the information retrieval (IR) community for many years [28]. This section surveys the literature of information retrieval through Direct Information Retrieval technique also known as keyword spotting technique.

Keyword spotting approaches are broadly classified as Character based keyword spotting & Word based spotting and sometimes there is an integration of the two. In character based spotting, features are represented at character level. Each character is encoded with a character shape code and they are joined together to represent a word. In Word based spotting, features are extracted at the word level directly instead of character level, which is insensitive to character segmentation error.

To avoid the difficulties of separating touching adjacent characters in a word image, segmentation-free methods have been researched in some document image retrieval systems, in which image matching at the word level is commonly utilized [17]. A segmentation-free word image matching approach treats each word object as a single, indivisible entity, and attempts to recognize it using features of the word as a whole.

Word level feature representation & matching can be performed either at the pixel level or feature level. Also pixel level processing is language independent and feature level processing is language dependent. Pixel level representation & matching has been performed by Statistical methods and Coarse feature based methods whereas the feature level representation has been addressed by Primitive String technique and Geometric Feature Graph (GFG) Extraction technique.

Even though, statistical models [5][6][7] identify keywords, this could work as a searching mechanism for the most likely sequence of characters, when a sequence of observations has been extracted from the input image. These methods can process only predefined keywords and a pretraining procedure is mandatory. A lexicon must be an available tool for resolving the ambiguity.

Initially, pixel based coarse features was addressed in Telugu document images [12], for information retrieval. Three coarse features such as word profiles, structural features and transform domain represented was addressed to indicate the features of a word image[13][3]. Feature values are normalized such that the word representations become insensitive to variations in size, fonts and degradations. Similarity between the words is identified using structural similarity of word images obtained by comparing the shapes and a sequence alignment score computed by Dynamic Time Warping procedure [21].

Pixel based coarse feature methods addressed for word level feature representation are language independent techniques and could work well across different languages. But a promising precision and recall cannot be achieved over a large corpus, since they require appropriate training sets. Due to non-language specific feature, pixel level feature vectors are very difficult to get indexed. On the other hand, feature based methods are advantageous, that they are easy to form query, easy to index and training free.

Primitive String generation for word images is a variation of character shape analysis. Character codes are not assigned to the characters individually. Instead, features are represented at the word level for image objects. In primitive string generation, an initial attempt was made by Yue lu and Tan to represent word images by a feature string as a whole [18], by scanning the word image column by column and giving a feature code for each column. They used Vertical Strokes, Long and short vertical strokes, Upper long vertical strokes and lower long vertical strokes as features to represent feature codes. Search process is initiated over it by synthesizing the feature string for the user specified query word according to the character sequence. Yue Lu extended the above [17], to accommodate many font faces and styles using a a Left to Right Primitive String (LPRS).

To simplify primitive string generation, a new word shape coding scheme has been proposed [29] to capture the document content through annotating each word image by a word shape code. In particular, word images were annotated by using a set of topological shape features including character ascenders/descenders, character holes, and character water reservoirs. With the annotated word shape codes, document images could be retrieved by either query keywords or a query document image. Primitive String generation technique is insufficient to represent Devnagari script since it mainly concentrates the characteristics of Roman script. This could not exploit the structural characteristics of Devnagari script. Therefore Geometric Feature GFG technique was introduced [9][10][11] to represent the features at word level in Hindi images.

However, feature extraction techniques discussed above are specific and language dependent. Consequently, for Tamil document images, need for information retrieval arises in the context of digitizing Tamil documents from ancient and old era to the latest. No specific word spotting technique reported above could be applied to the Tamil language directly since the features of Tamil language (i.e) shape of the Tamil characters, vary from other languages.

Tamil text recognition systems developed so far have their own constraints over the font faces or sizes of letters. Tamil text recognition systems are either restricted to a particular size or a range of font sizes. In addition, these text recognition systems [30] [31] suffer weaknesses to discriminate a group of closely resembling characters in the character set [32]. This also necessitates post processing or spell check to correct the errors occurred in recognition. Information retrieval would lead to a poor performance in these systems since spell check is essential after recognition.

These outstanding problems in Tamil text recognition systems motivated researches to the idea of developing LR-TB-FS technique [1], to retrieve information from Tamil document images. Here, the idea is based on the assumption that the technique devised would extract features of word images across various font faces and font sizes instead of training the shapes of the characters.

3. ONLINE INFORMATION RETRIEVAL SYSTEM

This section specifies the architecture of the online information retrieval system as depicted in Figure 1 and details the subsections involved in the system.

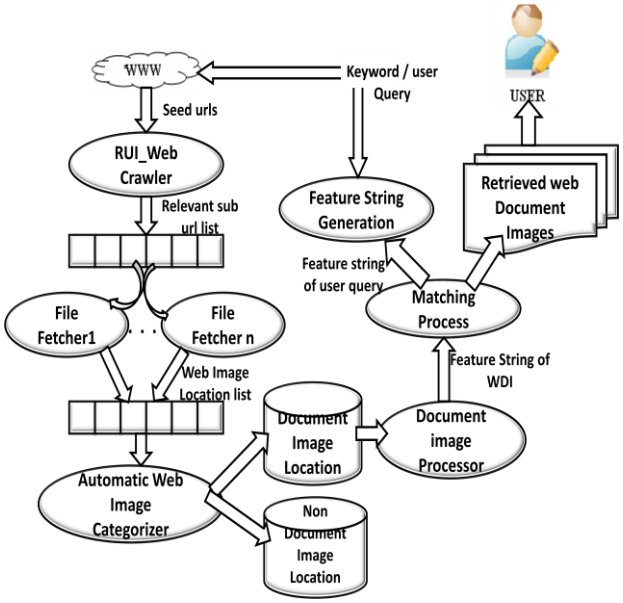

Fig 1: Online Information Retrieval Architecture

The basic processes involved in this online information retrieval system are:

- Seed Extractor.
- Web Image Files Grabber.
  - Relavant URL Identification(RUI) Web crawler
  - File Fetcher.
- Automatic Web Image Categorizer(AWIC)
- Document Image Processor
  - Preprocessing
  - Feature Extraction
  - Feature String Generation.
- Matching process.

Initially, seed urls are extracted from various search engines API or from experts [12]. After allowability checking, the allowable seed url's undergo RUI Web Crawling process to identify relavant sub urls. Filtering process filters out the visited and unallowable urls. Depending on the contents of relavant sub urls, the location of image files are identified. Subsequently, images are categorized and document images are discriminated from non document images.

Once the document images are categorized, Document Image Processing has been performed [1]. During this process, document images undergo preprocessing, Feature Extraction and feature strings are generated from the word images based on the statistical features of character shapes. This technique also provides the ability of matching textual query word given by the user with the imaged words of document images. First, feature strings represent word images in the documents. Later, A feature string matching method is then utilized to evaluate the similarity between the two feature strings (i.e) the query string provided by the user and the feature string generated from the word images in the document. If match exists, images and corresponding locations were presented to the user. This process is represented in Figure1.

3.1. Automatic Web Image Categorization

Web Image Categorization consists of the following processes such as Seed Extractor, Web Image Files Grabber, RUI Crawler and Categorizer.

3.1.1. Seed Extractor

Initially, a Seed Finder is used to find the web sites containing given topic related documents or which could lead to the web sites containing such document image files. Since it is critical to select the seed web page with knowledge acquirement, this is done through Google API or AltaVista or suggestions from the experts. These Seed urls act as the starting point for the entire crawling process.

3.1.2. Web Image File Grabber

- RUI Web Crawler.
- File Fetcher.

3.1.2.1 RUI (Relavant URL Identification) Web Crawler

Here, Web Crawler crawls and identifies relevant sub url list from the seed url set. Once the seed url is provided, all the sub url's starting from the seed url and all of its relevant links would be crawled and populated in a queue. Once the seed url gets exhausted, first url from the queue has been crawled. This process is done recursively, until no url exists in the queue. RUI Web Crawler process has been summarized in the following algorithm:

Input : Seed URL, Keywords.
Output : Relevant Sub URL's.

Process:

1. Begin RUI_WebCrawler
2. Get robots.txt for each seed url.
   i. If file exists and has "Disallow" statement, then
      Add to unallowable url_list.
   ii. Else
      Add to allowable_url_list.
3. For each url in (allowable_url_list)
   Enqueue (allowable_url_list,starting_url);
4. Next.
5. While (#url(allowable_url_list>0))
   a. url:=dequeueurl_with (allowable_url_list)
   b. page:=crawled_pagecontent (url);
   c. enqueue (crawled_url_list, (url, page));
   d. sub_url_list = extract_urls (page);
   e. For each page p in crawled_pagecontent
      i. If [page p has similarity with the keyword w in body or in title]
      Enqueue (relavant_sub_url_list)
   f. Next.
6. End While.
7. End RUI_Webcrawler.

In the above algorithm, operation Enqueue only adds a new URL to the queue if it is not already there. Operation Dequeue return the front element in queue and remove it out of queue. Operation extract_url(page), extracts all link context of url in the web page. Operation crawled_pagecontent (url), extracts the page content of the given url.

3.1.2.2 File Fetcher

File Fetcher uses sub url links discovered by RUI_Web Crawler to identify the image files. The image files are extracted by parsing through each sub url's page content. The file name has been parsed using the "img=" and "src" keywords. After identifying the image file name the path of the image location has also been identified by merging the

parent url and image file name. This process is carried out to eliminate the duplicated files and undesired file names. Later locations of image are stored in Vector list for further image categorization process. File fetching process has been summarized in the following algorithm:

Input : Relavant sub urls.
Output : Image File Locations.

1. Begin Image_File_Fetch()
2. Dequeue (Relavant_sub_url_list, sub_url);
3. for each (#sub_url)
   a. page:=crawled_pagecontent(sub_url);
   b. Identify file name between the clue words available in page p.
   c. Merge filename with parent url to find web_image_file_location.
4. Next.
5. End Image_File_Fetch.

3.1.3. Automatic Web image Categorization

Once the image files are fetched, this module tries to discriminate the document images from non document images. Discrimination of document images takes place by calculating the Energy, Entropy, Skewness and Kurtosis values of the web images as shown in the equations 1 to 4.

First parameter, Entropy measures the average bits per pixel. Small entropy indicates the presence of homogeneous regions in the image. Therefore, Document images tend to have smaller entropy than other type of images. Second Parameter, Energy is higher for homogeneous images and document images are generally characterized by large energy values. Third Parameter, Skewness is the degree of asymmetry of a histogram around the mean value and fourth one Kurtosis measures the peakness and flatness of the histogram.

$$\text{Energy} = \Sigma i\ hi\ (\log_2 hi) \quad \text{----------(1)}$$
$$\text{Entropy} = \Sigma i\ hi^2 \quad \text{----------(2)}$$
$$\text{Skewness} = \Sigma i((i-\mu)/\sigma)^3 hi \quad \text{----------(3)}$$
$$\text{Kurtosis} = \Sigma i((i-\mu)/\sigma)^4 hi - 3. \quad \text{----------(4)}$$

Where 'hi' is the value of i^{th} histogram bin. ($i \in [0\text{-}255]$ for all histograms)

Based on these four parameters (threshold range has been set for both kinds of images), images are classified as document and non document images as shown in the following algorithm:

1. Process AWIC
2. Dequeue (sub_url_list)
3. for each (web_image_file)
   a. Calculate Energy, Entropy for each web_images.
   b. If energy> threshold then and entropy < threshold
      Set as document image.
   c. Else
      Set as Non Document image.
   d. Scan the binarized image in horizontal direction.
   e. Calculate Average Black-to-White transition count for each row.
   f. If count> threshold then
      Set as Document image.
   g. Else
      Set as Non Document Image.
   h. If kurtosis< threshold and skewness lies between threshold
      Set as Document image
   i. else
      Set as non document image.
4. Next.
5. End AWIC process.

3.2. Document Image Processing

To process the document images, feature string generation technique, LR-TB-FS technique [1] has been employed here to generate feature strings for every word image. This contains four sub processes such as Preprocessing, Word Image Segmentation, Feature String generation, and Query Processing.

3.2.1. Preprocessing

As a preprocessing stage, image Binarization has been done. Binarization is a technique by which the grey scale images are converted to binary images. The most common method is to select a proper threshold for the image and then convert all the intensity values above the threshold intensity to one intensity value representing either 'black' or 'white' value. All intensity values below a threshold are converted to one intensity level and intensities higher than this threshold are converted to the other chosen intensity.

3.2.2. Word Image Segmentation

In this phase, line images are segmented through Horizontal Projection profile and word images are segmented from the line through Vertical Projection profiles.

Input : Binarized image.
Output : Segmented Word Images.

Process Logic:

- Lines are segmented using horizontal projection.
- Words are segmented using vertical projection.
- Characters are segmented using vertical projection.

3.2.3. Feature String generation

Once the word image objects are extracted from document images, its feature string represents them. The methodology employed here to represent word images is Left-to-Right, Top-to-Bottom Feature String (LR-TB FS), a code string framed from the Left to Right scan and Top to Bottom scan of a word image [1]. Line features and Transition features are used here to extract the primitives of a word image.

Each entity, called a primitive here, goes off vertical and horizontal scanning to define attributes for them [1]. Here primitive P is described using six attributes as represented in Equation (5)

$$P = (Vl, Hl, Vt, Ht1, Ht2, Ht3) \quad \text{----------(5)}$$

Vl	Vertical line count
Hl	Horizontal line count
Vt	Vertical Transition rate
Ht1	Ascender Zone Horizontal Transition rate
Ht2	Middle Zone Horizontal Transition rate
Ht3	Descender Zone Horizontal Transition

In Feature Extraction, number of vertical lines is detected through Vertical Line detection. Vertical Line detection technique detects the total number of vertical columns, left and right boundary pair of each vertical column consisting of consecutive black pixels along the vertical direction for $\theta = 90$ degrees along the height h of the entity e in the word image w. Number of horizontal lines is detected through Horizontal line detection. This technique detects the number of horizontal rows, top and bottom boundary pair of each row with

consecutive black pixels along the horizontal direction for θ = 0 degrees along the width of the entity e in the word image w. For every pair, if the width of the detected horizontal line is nearer to the threshold, it is assumed as horizontal line and the number of horizontal line count is incremented by 1.

After line feature extraction, transition features are extracted from the entity. Transition feature observes the number of black to white & vice versa disposition of the entities both vertically as well as horizontally through Vertical Transition and Horizontal Transition rate. Vertical Transition rate detects the vertical transition rate, by recording the black to white movements vertically in the Centroid area along the height *h* of the *entity e* in the word image *w*. Centroid area is defined as the middle third area over the width of the *entity e* along *h*. For every black to white and white to black disposition along the Centroid area, transition rate is recorded by incrementing 1.

Horizontal transition rate technique detects the horizontal transition rate, by recording the black-white disposition rate in ascender (above x-line), middle (between x-line and baseline) and descender zone (below the baseline) along the height *h* of the *entity e* in the word image *w*. For every black to white and white to black disposition over these zones, transition rate is recorded by incrementing 1.The attributes *ht1*, *ht2*, *ht3* represents the total number of black-white-black disposition along horizontal direction in ascender, middle and descender zones respectively. Feature String generation takes place as summarized below:

Input : Word image object.
Output : Feature String.

Process Logic:

- No of vertical lines are counted using vertical line identification.
- No of horizontal lines is calculated using horizontal line identification.
- Vertical transition rate was calculated.
- Horizontal transition rate was calculated in the ascender, middle, descended zones.
- Primitive P representation with six attributes calculated.

3.2.4. Matching process

For matching process, feature string for the query word has been synthesized from the primitive string of feature string table for Tamil language [1]. Feature String table consists of a primitive string for each character in Tamil language. Feature String for the query word can be generated by synthesizing the primitive string token of each characters in the word from the predefined table [1] and inserting a special character "c" among them to identify a spacing gap and "cw" to identify the end of primitives. Later, feature strings of query word are searched with the feature strings generated for the word images in the processed web document image. If the keyword gets a match with the feature string of the processed web document image, then the corresponding image and its location is shown to the user as a result of search process. This algorithm is summarized below

Input : Feature Strings of word images in Processed Web document image and Query word.
Output : Matched Web document images and its corresponding url address.

Process Logic:

- Query words are matched against the Feature String representations.
- Matched web image locations and images are stored in a Vector list to present to the users.

4. IMPLEMENTATION AND RESULTS

The seed urls and keyword were given as input for RUI_Web crawling process.

Figure 2 shows the result of RUI_Web crawling process, which is the list of relavant sub urls for the seed url http://www.palani.org. This process filters out visited suburls and irrelevant sub urls. The relavant sub urls alone stored in a Vector list.

After Image File fetching process the source file (.jpg) has extracted and the full image path has been identified. Figure 3shows the Web Image File Grabbing process. It shows the list of source file names and the absolute path of image locations list from the seed url http://www.palani.org.

After identifying the web image locations the document image categorization process has been performed. Based on the Energy, Entropy, skewness and Kurtosis, Web images are categorized. Figure 4 shows the categorized Document image. As the calculated values(energy, Entropy, BWR, etc.,) are compared against the threshold value this image has been categorized into Document image.

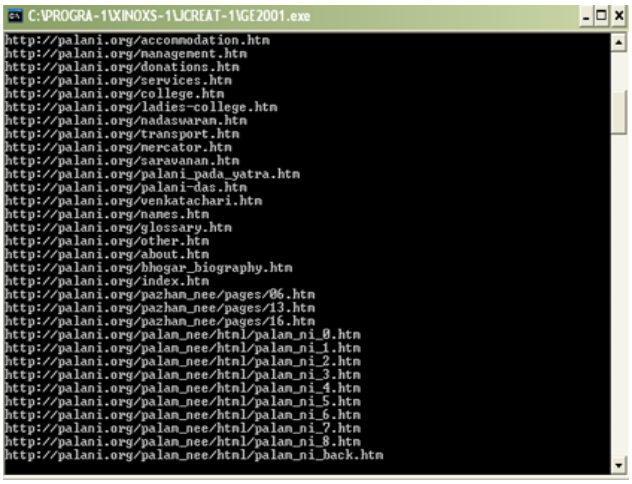

Fig: 2 Relavant sub url list after RUI_webcrawling

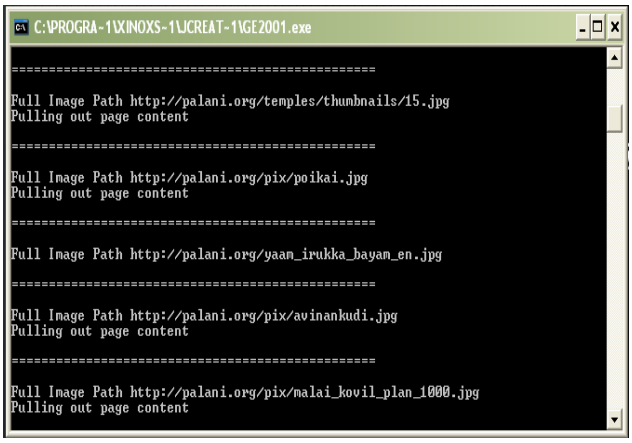

Fig: 3 Web Image File Fetching process

Figure 5 shows the categorized non document Image using Automatic Image categorization algorithm. The calculated values are compared against the threshold values and this image comes under Document image category.

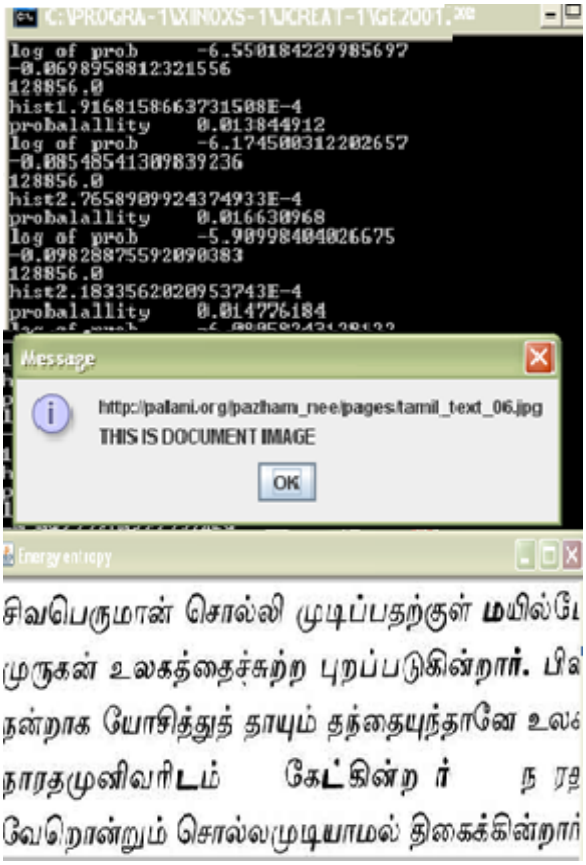

Fig: 4 Automatic Web image categorization- Categorized Document image

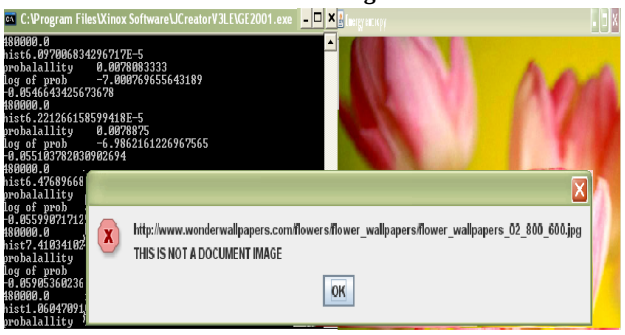

Fig: 5 Automatic Web image categorization-Categorized Non Document image

After the categorization of web images, each web document image was converted into its feature string representation and stored in a temporary file for matching progress. Figure 6 shows the feature string representation of the web document image.

.

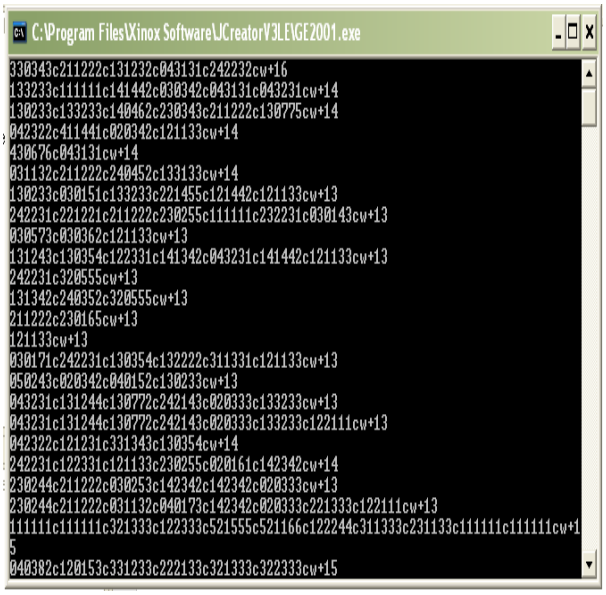

Fig: 6 Feature String Representation of Web document image

After getting the query word from the user it is converted into a feature string. Figure 7 shows the GUI which gets the Tamil query word from user and the feature string representation generated for the query word.

Feature String generated for the query word has been compared against each and every feature strings generated from the web document images. If match exists, the retrieved image along with its image location is presented to the user as shown in figure 8

Fig: 7 Feature String Representation of User query

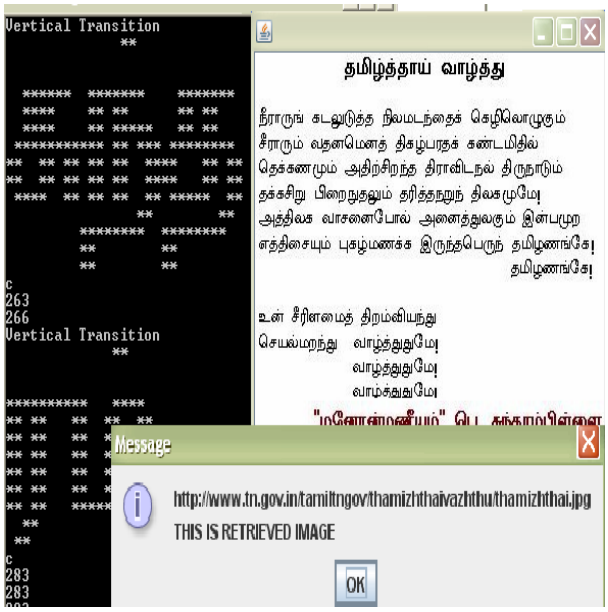

Fig: 8 Retrieved Web document image and its location.

5. PERFORMANCE ANALYSIS

This section explains the performance analysis of the Automatic Web image Categorization algorithm and Information Retrieval from categorized web document images.

Table 1 shows the performance evaluation for the given seed url's and all the images grabbed from the websites resulting from the seed url's. This evaluation also shows the performance of AWIC in the categorization rate of images available in the web sites and Table 2 shows the accuracy rate of categorization for the seed url's. Table 3 shows the results obtained for sample query words. Figure 9 shows the categorization accuracy obtained for Document and Non document images. Figure 10 depicts the precision and recall rate obtained for sample query words. Recall and precision for the query words were computed. Percentage of relevant words that are retrieved from the entire collection is represented as recall; whereas percentage of correctly retrieved words, from retrieved collection, is represented as precision.

TI – Total images in sub Url's
PI – Processed Images
TND – Total number of Non Document Images
TD – Total number of Document Images
CND – Categorized Non Document Images
CD – Categorized Document Images

Tab: 1 Categorization results of AWIC

SEED URLS	TI	PI	TND	CND	TD	CD
www.bhc.edu.in	144	144	135	135	9	8
www.biblelamp.in	35	35	31	30	4	3
www.palani.org	395	395	379	376	16	16
www.tamilinayam.com	80	80	48	48	32	27
www.jesusredeems.com	113	113	96	96	17	12
www.tnou.ac.in	92	92	79	79	13	12
www.tvuni.in	15	15	3	3	12	10
www.b-u.ac.in	116	116	84	84	32	32
www.alagappauniversity.ac.in	87	87	87	87	0	0
www.tn.gov.in	48	48	37	35	9	9
www.thisgospelthisgod.com	21	21	0	0	21	21

Tab: 2 Accuracy rate of AWIC for seed url's

SEED URLS	TP	TN	FP	FN	ACC	Sensitivity	Specificity
www.bhc.edu.in	135	8	0	1	99.3	99.26	100
www.biblelamp.in	30	3	1	1	94.28	96.77	75
www.palani.org	376	16	3	0	99.24	100	84.21
www.tamilinayam.com	48	27	0	5	93.75	90.56	100
www.jesusredeems.com	96	12	0	5	99.08	95.04	100
www.tnou.ac.in	79	12	0	1	98.91	98.75	100
www.tvuni.in	3	10	0	2	86.66	60	100
www.b-u.ac.in	84	32	0	0	100	100	100
www.alagappauniversity.ac.in	87	0	0	0	100	100	0
www.tn.gov.in	35	9	2	0	98.56	100	81.81
www.thisgospelthisgod.com	0	21	0	0	100	0	100

Tab: 3 Results of the images

Query word	Total words in url	Processed words	Retrieved words
Query 1	3	3	2
Query 2	1	1	1
Query 3	1	1	1
Query 4	2	2	1
Query 5	1	1	1
Query 6	1	1	1
Query 7	1	1	1
Query 8	1	1	1
Query 9	2	2	2
Query 10	1	1	1

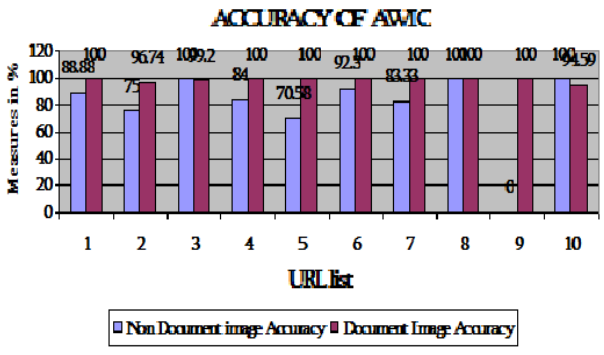

Fig: 9 Accuracy of AWIC Process for sample query words

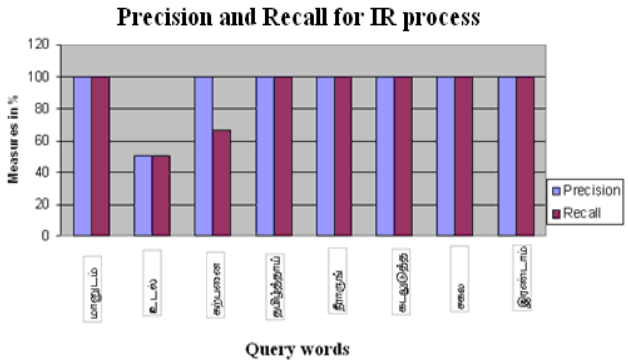

Fig: 10 Precision and Recall for Query words

6. CONCLUSION AND FUTURE WORK

This system provides a framework for the development of an Information Retrieval system for the Tamil document images in WWW. Initially, this system concentrates on the Automatic Image categorization process over web images by employing a filtering technique to discriminate the document images, available in WWW. The strength of this technique lies in capturing the image information by intensity and frequency histograms for discrimination of web document images. Another major benefit of this system is that, during IR it does not convert the document image into text, but analyses the basic characteristics or shapes and represents the word image as feature string. After generating feature strings from the Tamil Web document images, keyword-based IR is performed to display the relevant results (i.e. relevant images) to the user.

As a future work, this system could be extended for bilingual and Multilingual IR from web document images. Further, this system could be improved by adding additional parameters to

resolve the problem of retrieving background information from web document images, where pictures and text are intermingled with each other.

7. REFERENCES

[1] Abirami .S, Manjula.D, "Feature string-based intelligent information retrieval from Tamil document images", International Journal of Computer Applications in Technology, 2009, Vol. 35, Nos. 2/3/4, pp 150-165.Ding, W. and Marchionini, G. 1997 A Study on Video Browsing Strategies. Technical Report. University of Maryland at College Park.

[2] Arvind Arasu, Junghoo Cho, Hector Garcia-Molina, Andreas Paepcke, and Sriram Raghavan, "Searching the Web", ACM Transactions on Internet Technology, 2002, Vol.1, No.1,pp 2-43.

[3] Balasubramanian A., Meshesha M. and Jawahar C.V., 'Retrieval from document image collections', Proceedings of the International Workshop on Document Analysis Systems, LNCS 3872, 2006, pp.1-12.

[4] Chakrabarti, Van den Berg, and Dom. "Focused crawling: a new approach to topic-specific Web resource discovery", In Proceedings of the 8th International World Wide Web Conference, 1999.

[5] Chen F.R., Wilcox L.D. and Bloomberg D.S., 'Detecting and locating partially specified keywords in scanned images using hidden markov models', Proceedings of the International conference on Document Analysis and Recognition, 1993, pp.133-138.

[6] Chen F.R., Wilcox L.D. and Bloomberg D.S., 'A comparison of discrete and continuous hidden markov models for phrase spotting in text images', Proceedings of the International conference on Document Analysis and Recognition, 1995, pp.398-402.

[7] Chen F.R. and Bloomberg D.S., 'Extraction of thematically relevant text from images', Symposium on Document Analysis and Information Retrieval, 1996, pp.163-178.

[8] Diligenti.M, Coetzee.F.M, Lawrence, Giles, and Gori. "Focused crawling using context graphs." In Proceedings of the 26th International Conference on Very Large Data Bases, 2000.

[9] Harit G., Chaudhury S., Gupta P., Vohra N. and Joshi S.D., 'Model guided Document Image Analysis system', Proceedings of the Sixth International Conference on Document Analysis and Recognition, 2001, pp. 1137-1141.

[10] Harit G., Chaudhury S. and Paranjpe J., 'Ontology guided Access to Document Images', Proceedings of the Eighth International Conference on Document Analysis and Recognition, 2005, pp. 292-296.

[11] Harit G., Garg R. and Chaudhury S., 'An integrated scheme for compression and interactive access to document images', Proceedings of the International conference on Computing: Theory and Applications, 2007, pp. 506-511.

[12] Jawahar C.V., Meshesha M. and Balasubramanian A., 'Searching in Document Images', Proceedings of the International conference on Visualization, Graphics and Image Processing 2004,pp. 622-627.

[13] Jawahar C.V., Million M. and Balasubramanian A., 'Word level access to Document Image Datasets', Proceedings of the Workshop on Computer Vision, Graphics and Image Processing,2004,pp. 73-76.

[14] Jung.G.S and Gudivada, "Autonomous tools for information discovery in the world-wide web," School of Electrical Engineering and Computer Science, 1995.

[15] Jorgensen, "Attributes of Images in Describing Tasks," Information Processing and Management, Vol. 34, nos. 2–3, 1998, pp. 161–174.

[16] Kompatsiaris, Triantafyllou and Strintzis M.G., "A World Wide WebRegion-Based Image Search Engine," 11th International Conference on Image Analysis and Processing (ICIAP'01), 2001.

[17] Lu.Y and Tan.C.L ,"Information Retrieval in Document Image Databases", IEEE Transactions On Knowledge And Data Engineering, 2004, Vol. 16, No. 11, pp.1398-1401.

[18] Lu Y. and Tan C.L., 'Word Searching in Document Images Using Word Portion Matching', Document Analysis Systems V, Lecture Notes on Computer science, *2002*, Vol. 2423, pp.319-328.

[19] Najork and Wiener.L.N. "Breadth-first search crawling yields high quality pages." In Proceedings of the 10th International World Wide Web Conference, 2001.

[20] Neil C. Rowe, "Marie-4: A High-Recall, Self-Improving Web Crawler That Finds Images Using Captions ", IEEE Intelligent Systems Archive, 2002, Vol.17 , No.4 , pp: 8 – 14.

[21] Rath T. and Manmatha R., 'Features for word spotting in historical manuscripts', International conference on Document Analysis and Recognition, 2003, pp. 218-222.

[22] Sclaro, "World wide web image search engines," in NSF Workshop on Visual Information Management, Cambridge, MA, June 1995.

[23] Shen Jin-Xing, "An ontology-based adaptive topical crawling algorithm", 2008.

[24] Smith.J.R and Chang, "Visually searching the Web for Content," IEEE Multimedia Magazine, 1997, Vol. 4, no. 3, pp.12-20.

[25] Smith S. F. and Chang "An Image and Video SearchEngine for the World-Wide Web", Proceedings of IS&T/SPIE , Storage & Retrieval for Image and Video Databases, 1997.