Information Extraction among Scanned Document Images in Database

[1] Rashmi M. Choudhari, [2] Dr. D. M. Dakhane

[1]Student, [2] Professor

[1] Department of Computer Science and Engineering,

[1] Sipna College of Engineering, Amravati, India

Abstract: Information extraction is a key feature for mining any data so the detection of duplicate images is a useful means of indexing a large database of documents. An algorithm for duplicate document detection is proposed in this project that operates directly on images that have been symbolically compressed using techniques related to the ongoing JBIG2 standardization effort. This report describes an optical character recognition (OCR) method that recognizes the text in an image by deciphering data from the compressed representation. It recognizes the text in an image by deciphering the sequence of occurrence of blobs in the compressed representation. We propose a Hidden Markov Model (HMM) method for solving such deciphering problems and suggest applications in multilingual document duplicate detection. It is observed that it can recover better than 90% of the text in compressed document images and that this is sufficient to identify duplicates in a large database.

Index Terms - **Document Image, Duplicate detection, IM^3, Optical Character Recognition, Symbolic Compression.**

I. INTRODUCTION

Lost documents are a significant problem for office workers. A recent study estimates that at any time 3% to 5% of documents are lost and the average cost of lost documents to a Fortune 500 company is in the range of $3 million to $5 million [10]. An obvious solution to this problem is for users to scan all their documents. However, for any number of reasons, not the least of which are users' natural reluctance to alter their work practices, such an approach has not been widely adopted [3]. In fact, recent results indicate that for new retrieval techniques to gain wide acceptance they should be easy to use, be familiar and require as little user effort as possible.

Document matching is an important component of a document image storage system, allowing for removal of duplicates, copyright violation detection and other applications. Since document images are usually stored and transmitted in compressed formats, considerable advantages are realized by performing the matching process directly on compressed images [5][10]. One technical challenge is the extraction of meaningful information from compressed data.

Duplicate detection is particularly important for large document databases like that produced by the Infinite Memory Multifunction Machine (IM) .The IM^3 system (Hull and Hart, 1998; Hull et al., 1999) captures a copy of every printed, copied, or faxed document generated in an office. This guarantees that almost any document a user might need will be available when they need it. This concept was developed after it was observed that even though the obvious method for document capture, namely scanners, were commonly available, they were not commonly used.

The dramatic increase in the use of document images in recent years has led to research in compression technology for textual images. Symbolic compression schemes preserve much of the structure in a document image thereby facilitating feature extraction. They cluster individual blobs in a document and store the sequence of occurrence of clusters and representative blob templates. This kind of compression scheme was originally proposed binary images of text [1]. Lossless compression algorithms can be designed around this idea by supplementing a coding scheme for the residuals that result from pattern matching [11]. It has been shown that such separate coding of patterns and residuals achieves better compression ratios than conventional methods. Numerous algorithms based on the pattern matching approach such as JBIG2 [4] and others [8][7] have been proposed recently.

Economic considerations are always important factors when users decide whether to adopt new technologies. An important consideration in the design of the IM^3 system was the relative cost of printing a document on seminar vs. storing an image of the same document on magnetic disk. Of course, there is a wide variation in the price of seminar and toner needed to print the range of documents encountered in the typical office. For the purposes of this analysis, it was assumed that, on average, the cost of an 8.5x11 inch sheet of seminar is one cent. It was also assumed that a 400 dpi binary image of the same document on average would require 100 KB. At the time the IM^3 project was started (late 1993) it was observed that it cost about 3 cents for 100 KB of magnetic disk storage. This was just for the disk space. It did not take into account the cost of the computer, etc. However, we projected that over time these costs would significantly decrease and eventually become less than the cost of a sheet of seminar. That time arrived sometime in 1996. Today, it costs about 0.27 cents for 100 KB of disk space. This is significantly less than the cost of a sheet of seminar. The 4:1 difference in cost of the two media now favors the adoption of a document storage and retrieval system like the IM^3.

A. IM^3 System Design

Documents stored in the IM^3 are accessed with a web browser. Each user has a home page that provides a portal to their document collection. A number of techniques are provided for search and retrieval. These include full text search and various methods for browsing based on the dates when documents were captured. A method for duplicate detection would give users a means for retrieving exact copies and other versions of a given document. Version retrieval would be particularly useful when after retrieving a document by full text search with a certain set of keywords; the user would like to retrieve other versions of that document. Even though substantial amounts of text may be common between versions, they might not all share the keywords that were used for full text search.

Specially modified digital photocopiers were developed that automatically capture an image of every copied document. Aside from a user identifying himself by pressing a button on a touchscreen, this process is completely transparent. The captured images are transferred to the document server where they are permanently stored and indexed for later retrieval. Print jobs are automatically captured by software running on a Unix print server. A copy of every printed document is transferred to the document server as it is sent to a printer. This is done by a filter in the spooling system that is applicable to jobs printed on PC's, Apple computers, and Unix workstations. In this way the capture of printed documents is completely transparent to the user and is independent of any application software. Every document sent to printers serviced by the Unix server is saved.

An indexing process is applied to every saved document. The images from the photocopiers are OCR'd. Text is extracted from the postscript files for printed documents. This is used to choose keywords for each document and build data structures for full text retrieval. Thumbnail images are also calculated at several resolutions (4 dpi, 8dpi, and 72 dpi) for use in various brows able interfaces

II. LITERATURE SURVEY

R. N. Ascher and George Nagy, proposed a method of video compaction based on transmitting only the first instance of each class of digitized patterns is shown to yield a compaction ratio of 16:1 on a short passage of text from the IEEE Spectrum. Refinements to extend the bandwidth reduction to 40:1 by relatively simple means are proposed but not demonstrated. Very high degree of bandwidth reduction on printed text digitized by an optical scanner may be achieved by transmitting or storing only the first instance of each pattern class and thereafter substituting this exemplar for every subsequent occurrence of the symbol. This process, which approaches in efficiency actual recognition of the text but does not require the assignment of correct alphanumeric labels to the characters. The determination of which characters are to be saved in video form is based on binary correlation. A character is saved (in our terminology, becomes a prototype) only if it is not highly correlated with any of the previously saved characters. If a character is not saved, then it is assigned to the prototype most highly correlated with it, and only the identification number of this prototype is transmitted or stored. The method is derived from a no supervised method of character recognition in which the prototypes are saved for subsequent identification by a human operator. In the procedure described here, however, the prototypes are either saved or transmitted without ever being assigned alphanumeric identities. The routines necessary to scan isolate, and correlate the characters are outlined, and are used without change in the present experiment. The correlative coding process does require a considerable amount of computation, but the decoding process-looking up the video pattern corresponding to a prototype number-is straightforward and rapid. Hence, this compaction method is particularly suitable for archival applications, in which the high cost of encoding is justified by the high degree of compaction achieved and the ease and faithfulness of reconstitution [1].

Richard G. Casey and George Nagy, an unconventional approach to character recognition is developed. The resulting system is based solely on the statistical properties of the language, therefore it can read printed text with no previous training or a priori information about the structure of the characters. The known letter-pair frequencies of the language are used to identify the printed symbols in the following manner. First, the scanned characters are partitioned into distinct groups of similar patterns by means of a distance measure. Each class (at most 26 are permitted) is assigned an arbitrary label, and an intermediate tape, containing these temporary labels of the symbols in the original sequence, is generated. In the second phase of the program, the matrix of bigram frequencies of the labels is compared to a frequency matrix obtained from a large sample of English text. The labels are then assigned alphabetic symbols in such a way that the correspondence between the two matrices is maximized. The method is tested on a 100 000-character data set comprising four markedly different fonts. The system to be described presents a striking contrast to the usual methods of character recognition. The processor is exposed to an unlabeled representation of each character in a passage of printed text, but it is given no information regarding the significance of the various elements in these patterns. Unlike conventional classifiers, it is never "trained" on identified samples. In principle, this processor has no stored data to aid it except contextual information in the form of a table of letter-pair frequencies derived from other samples of text [2].

Howard P, Kassentini F, Martins B, proposed the Joint Bi-level Image Experts Group (JBIG), an international study group aliated with iso/iec and itu - t, is in the process of drafting a new standard for lossy and lossless compression of bi-level images. The new standard, informally referred to as JBIG2, will support model-based coding for text and halftones to permit compression ratios up to three times those of existing standards for lossless compression. JBIG2 will also permit lossy preprocessing without specifying how it is to be done. In this case compression ratios up to eight times those of existing standards may be obtained with imperceptible loss of quality. It is expected that JBIG2 will become an International Standard by 2000. JBIG2 is an emerging iso/iec International Standard for lossy and lossless bi-level image compression. It is being drafted by the Joint Bi-level Image Experts Group (JBIG), a Collaborative Team" that reports both to iso/iec. As the result of a process that ended in 1993, JBIG as a Collaborative Interchange" produced a bi-level image coding standard formally designated itu - t Recommendation t.82 j International Standard iso/iec 11544, and informally known as JBIG or JBIG1. The authors of this paper are all active members of JBIG, although among us only Dr. Ono was involved in JBIG1 [4].

Jonathan J. Hull and Peter Hart, proposed a complete, working document storage and retrieval system is described. Designed to help users solve the problem of lost documents, this system illustrates the concepts of automatic document capture and easy retrieval. Every document a user copies or prints is automatically indexed and saved for later retrieval. A prototype implementation of such a system was constructed and used daily by approximately 20 people for over two years. Lost documents are a significant problem for office workers. A recent study estimates that at any time 3% to 5% of documents are lost and the average cost of lost documents to a Fortune 500 company is in the range of $3 million to $5 million [10]. An obvious solution to this problem is for users to scan all their documents. However, for any number of reasons, not the least of which are users' natural reluctance to alter their work practices, such an approach has not been widely adopted [3]. In fact, recent results indicate that for new retrieval techniques to gain wide acceptance they should be easy to use, be familiar and require as little user effort as possible. This paper proposes a system design called the Infinite Memory Multifunction Machine (IM3) in which every document a user copies, prints, or faxes is automatically captured and indexed for later retrieval with a web browser. The automatic capture process is performed as a natural side-effect of copying, printing, or faxing and is almost completely transparent to the user.

This removes any need for the user to decide at the time a document is processed whether it should be saved. By so doing, users are almost guaranteed that when they need to find a document, the system will contain a copy of it [6].

Jonathan J. Hull, Dar-Shyang Lee, John Cullen, and Peter Hart, The Infinite Memory Multifunction Machine (IM3) is a document storage and retrieval system that solves a large portion of the problem of lost documents [6]. It captures a copy of every printed, copied, or faxed document generated in an office. This guarantees that almost any document a user needs will be available when they need it. This concept was developed after it was observed that even though the obvious method for document capture, namely scanners, were commonly available, they were not commonly used. The design of the Infinite Memory Multifunction Machine (IM3) was described. This system solves a large portion of the problem of lost documents by capturing an electronic copy of every document that users copy, print, or fax. This effectively reduces the effort expended by the user at capture time to zero. However, this increases the effort that must be expended at retrieval time since users must filter through large numbers of documents. Two document analysis techniques were described that help users retrieve documents in such a system. One technique detects duplicate documents. Another method automatically files a document in an existing hierarchy. Experimental results demonstrated the efficacy of both methods [8].

III. IMPLEMENTATION

Duplicate documents can be a significant problem in large collections of database. Ideally document images are not identified while scanning a particular document. While scanning a document it will create an image of that document know as document image which is stored on a server (here IM3 server). So suppose in an office where large number of computers are connected and each scanning some documents, so there is a maximum chance of storing duplicates documents, which in terms uses large storage space and it leads into large database with duplicate document images.

Text from images can be extracted using Optical Character Recognition (OCR). OCR works in three phases as preprocessing, segmentation, character recognition. Preprocessing is the first phase which uses different techniques for making text easy to extract from images. In segmentation phase, each character is isolated. Then this will be given as input to OCR recognition phase which will compare it with training data-set and will recognize character

Classical ciphers are the earliest schemes of cryptography. There are usually two types of classical ciphers, namely transposition ciphers and substitution ciphers. Although, from the security point of view, classical ciphers are no match to the recently developed ciphers, they have not lost their importance because most of the commonly used modern ciphers use classical ciphers as their building blocks. In fact, several complex algorithms can be formed by mixing substitution and transposition operations. Modern block ciphers such as DES and AES iterate through several stages of substitution and transposition.

The main objectives are:

- To implement the system that converts the document image into text for better recognition.
- To apply OCR algorithm for extracting text from document image.
- To detect duplicates document image in server folder.
- To apply hash function algorithm to generate text impression to remove duplicates.

➢ **Architecture Block Diagram of proposed system:**

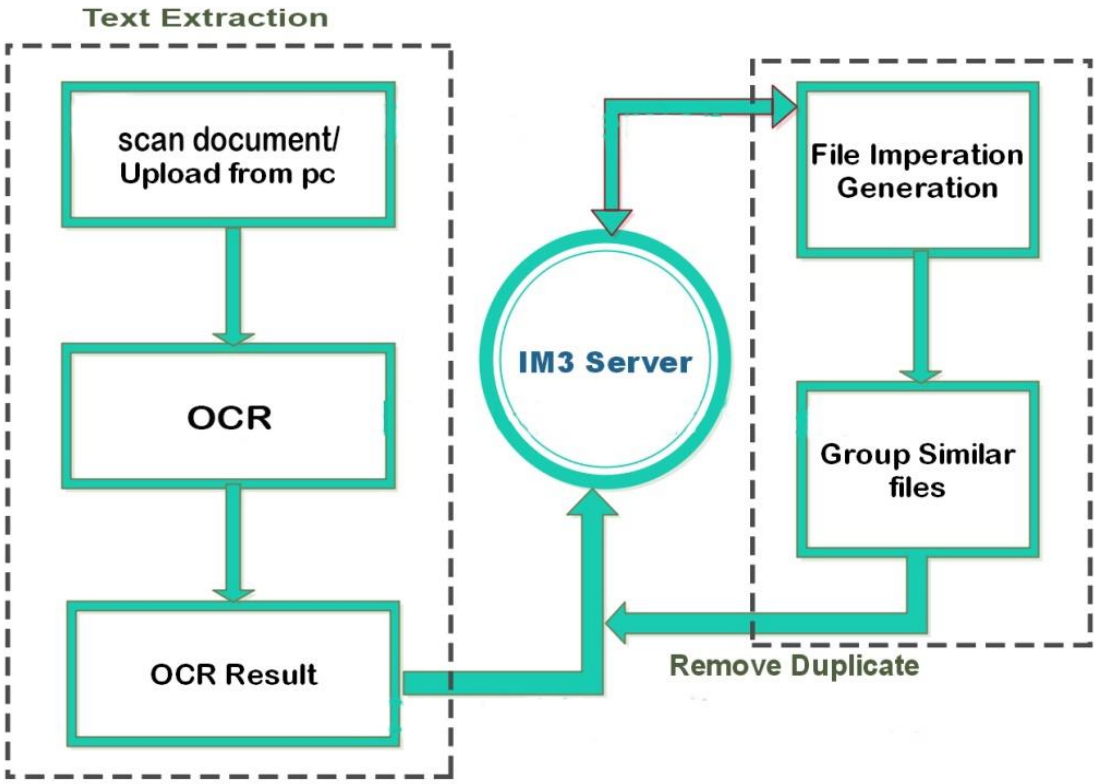

Figure 1: System Architecture

➢ **Description about blocks and connectivity between blocks**

- Scan document or upload from pc:

As this project completely works on document image, so first of all we need a document image. This block helps us to get the document image, in this block two options are provide to get document image. First, by scan the document at the run time i.e. by clicking on scan document button it will scan the document from the connected scanner. Second, by browsing the already scanned document image from the local system, here user is able to upload the scanned document image from his system to perform the OCR.

- OCR (Optical Character Recognition):

Optical character recognition is the mechanical or electronic conversion of images of typed, handwritten or printed text into machine-encoded text, whether from a scanned document, a photo of a document, a scene-photo (for example the text on signs and billboards in a landscape photo) or from subtitle text superimposed on an image. It is widely used as a form of information entry from printed paper data records, whether passport documents, invoices, bank statements, computerised receipts, business cards, mail, printouts of static-data, or any suitable documentation. It is a common method of digitising printed texts so that they can be electronically edited, searched, stored more compactly, displayed on-line, and used in machine processes such as cognitive computing, machine translation, (extracted) text-to-speech, key data and text mining. OCR is a field of research in pattern recognition, artificial intelligence and computer vision.

In this block the scanned document image is examined using OCR, the complete image is recognized to get the complete text from the image. After performing OCR the document image is converted into text and this text is saved with a .txt extension.

- OCR Result:

After the OCR is performed on document image, the converted result is stored on a server with the original document image. User can view the converted teat file by clicking OCR result. Here the server folder is treated as IM3 server where all the original and OCR document are stored.

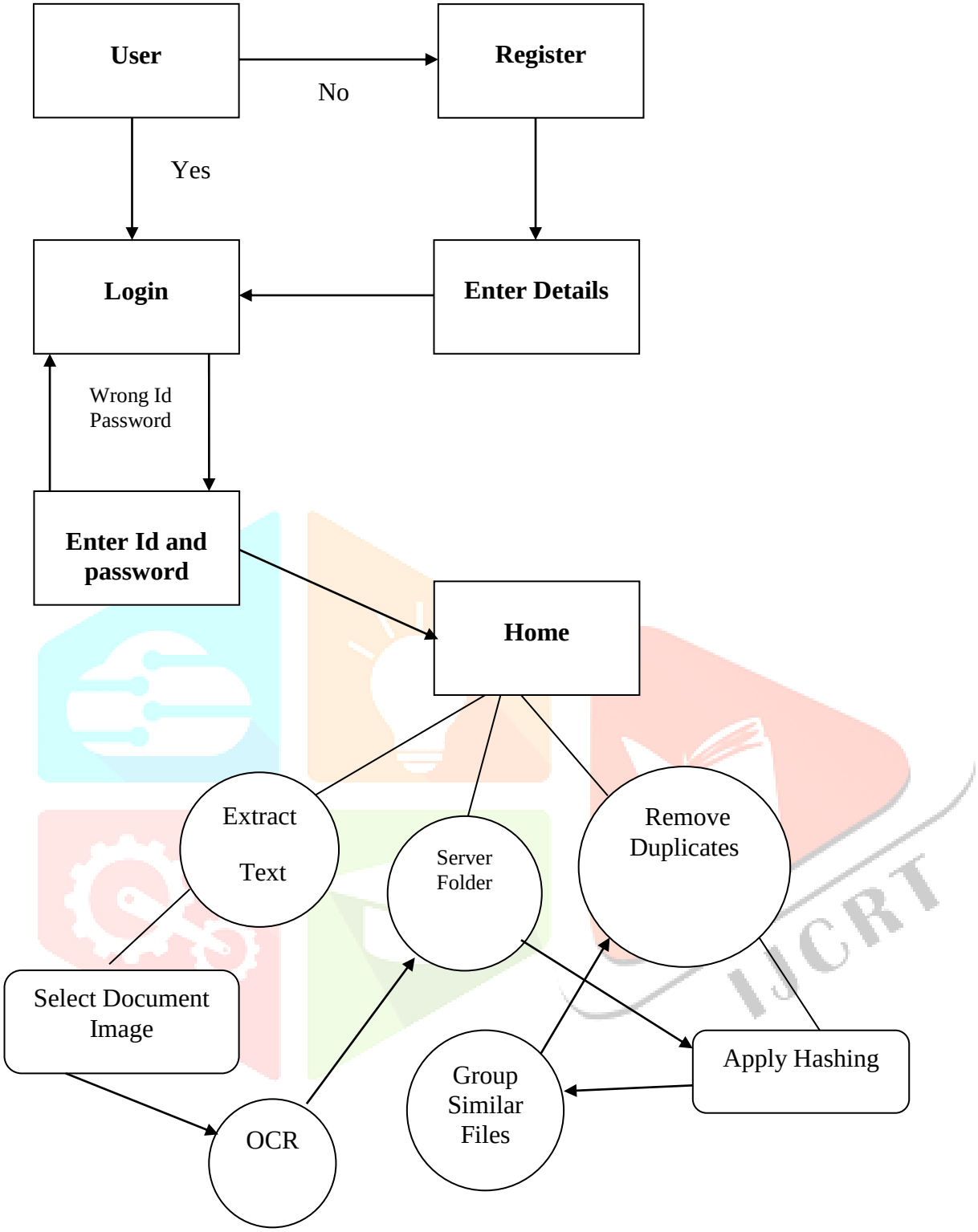

Figure 2: Data Flow Diagram level 2

I. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.

II. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.

III. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.

IV. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

- IM3 Server (Server Folder):

The Infinite Memory Multifunction Machine (IM3) is a document storage and retrieval system that solves a large portion of the problem of lost documents. It captures a copy of every printed, copied, or faxed document generated in an office. This guarantees that almost any document a user needs will be available when they need it. This concept was developed after it was observed that even though the obvious method for document capture, namely scanners, were commonly available, they were not commonly used. The IM3 makes document capture effortless by saving an electronic copy of every document that users copy, print, or fax. Furthermore, users are not asked whether any particular document should be captured no conscious decision is required at capture time. Thus, every person in an office that copies, prints, or faxes a document automatically contributes data to the IM3.

Initially the scanned document images are stored somewhere, here a folder is created to store all the scan documents images as well as the converted text file and this folder is treated as server folder.

- Remove Duplicates:

In this phase after the first phase text extraction is completed, this phase removes the duplicate files which are stored in server folder. As our main goal is to remove duplicate document images from server folder, here in this phase this task is accomplished. The converted text files which are stored in server folder are examined and group, this grouping of files are on the basis of hash function. This system generates a file impression code using hash function algorithm for all the files which are placed in server folder, if the two or more files are exactly same then the hash code is also same. On the basis of this hash code the system bundle the file with the same hash code into a group that means this group contents the duplicate files. After grouping user has the authority to view this file by clicking it as well as to remove the duplicate files by clicking delete button. So here by doing this the system removes all the duplicate documents images from the sever folder.

IV. STEP BY STEP PROCESS OF EXTRACTING DOCUMENT IMAGES

We assume that all document images are compressed with a ``symbolic'' technique (e.g., JBIG2 (Howard et al., 1998)). Features are extracted directly from the compressed version of document images [3]. A comparison procedure determines whether the feature descriptions of any two documents are similar enough for the original documents to be duplicates. Hull et al.introduced a Symbolic compression for binary document images first clusters connected components, which often correspond to isolated characters. A unique identifier is then assigned to each cluster. The compressed document contains one image for each cluster and the sequence of identifiers for the connected components (also called blobs) in the original image[7]. This sequence of identifiers corresponds to the sequence of occurrence of characters in the original document.

The extraction of information from compressed document images is useful since the compression algorithm not only reduces the size of the image, providing less data to process, but also represents characteristics of the original image in the compressed data stream that can be used directly to compute information about original document. CCITT groups 3 and 4 compression are one example. These methods include pass codes in the compressed data stream, which are attached to connected components. The configuration of pass codes in CCITT-compressed document images has been used for skew detection [27] and duplicate detection [12, 18] Symbolic compression has recently been proposed for inclusion in the JBIG2 standard [10]. Symbolic compression methods were first discussed by Ascher and Nagy [1]. More recent works include [9, 17, 28, 30]. In symbolic compression, images are coded with respect to a library of pattern templates. Templates in the library are typically derived by grouping (clustering) together connected components that have similar shapes. One template is chosen to represent each cluster. The connected components in the image are then stored as a sequence of template identifiers and their offsets from the previous component. In this way, an approximation of the original document is obtained without duplicating storage for similarly shaped connected components.

Minor differences between individual components and their representative templates, as well as all other components which are not encoded in this manner, are optionally coded as residuals. An example of symbolic compression is shown in Figure 5. After connected component clustering, the original document image is represented as a set of bitmap templates, “A a h i s t” in this example, their sequence of occurrence in the original image (“0 1 2 1 5 3 4 1 2 1 5 3 4 1 5 1 5”), as well as information about the relative geometric offset between adjacent connected components (e.g., (+2, 0) means the beginning of the second component in the sequence is 2 pixels to the right of the end of the first component), and a compressed residual image. The residual is the difference between the original image and the pattern templates. This data can be compressed with arithmetic coding or another technique. A lossy representation for a symbolically compressed image could be obtained by not storing the residual image. Symbolic compression techniques improve
compression efficiency by 50% to 100% in comparison to the commonly used Group 4 standard [19, 29]. A lossy version can achieve 4 to 10 times better compression efficiency than Group 4 [29].

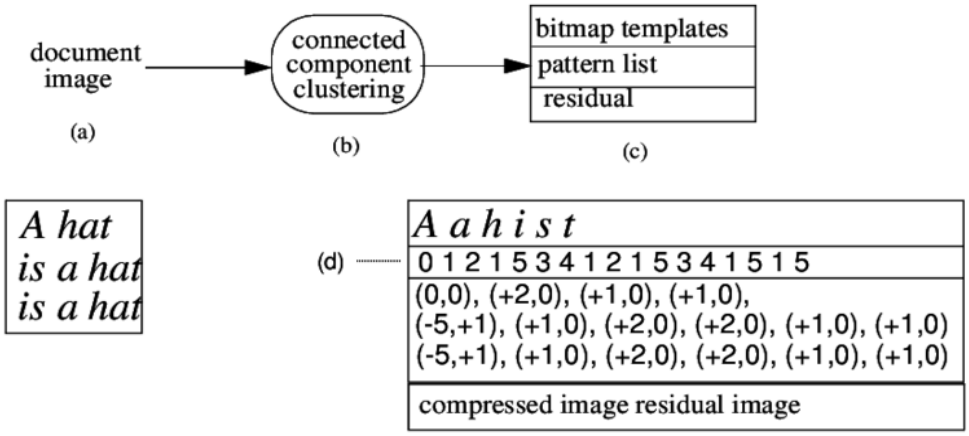

Figure 3: Example of Depiction of symbolic Compressed Images

An idealized example of a symbolically compressed (similar to JBIG2) document image is shown in Figure 3. An original document image is shown (a) as well as its compressed form (b). The unique letters in the original image are represented as individual sub-images and numeric identifiers at the top of (b). The sequence of identifiers shown in Fig. 3(c) encodes the order in which the corresponding sub-images occurred in the original image (a). For example, ``0 1 2 3" are the first four sub-images in this sequence. They correspond to the first four letters in the image, ``PALO". The x±y locations of the sub-images and image residual data are also encoded in the compressed format.

The characteristic of symbolic compression that we use for duplicate detection is the sequence of cluster identifiers (``0 1231243032567"in Fig. 3(c)). This sequence encodes a representation for the text in the original document. Since each cluster, for the most part, corresponds to a single character, we can treat the sequence of cluster identifiers as a substitution cipher.

A substitution cipher replaces one character with another to produce an enciphered message. The original plain text can be recovered by a deciphering algorithm that computes the pairwise correspondence between symbols in the enciphered message and plain text characters. For the example shown in Figure 3, this correspondence is f 0; P ; 1; A ; 2; L ; 3; O ; 4; T ; 5; I ; 6; C , 7; Y g.We apply a deciphering algorithm to the sequence of cluster identifiers. It computes a pairwise correspondence between cluster identifiers and letters. This correspondence is used to recover the text in the original document image. This is essentially OCR'ing the document without actually applying any OCR techniques. A similar idea was first proposed in (Casey and Nagy, 1968). Our method takes advantage of the image preprocessing done by the symbolic compression technique. Also, we developed a new algorithm for substitution cipher decoding that takes into account characteristics of symbolic clustering in document images. Other techniques for substitution cipher decoding are described elsewhere.

Rabiner, L.R., Juang, et al,[11] introduced the deciphering algorithm reads the sequence of cluster identifiers from a symbolically com pressed document image and uses character transition probabilities and a hidden Markov model (HMM) to estimate the text that appeared in the original document. There might not be a decision for every character and all the decisions might not be correct. However, enough of the text is usually correctly recovered that accurate duplicate detection can be performed.

The text strings extracted from two documents are compared using th conditional n-grams they have in common[15]. A conditional n-gram is a sequence of n characters where each character satisfies a predicate. This predicate converts the original document into a new string from which n gram indexing terms are extracted. For example, we used a predicate that every character must follow a space. Therefore, the new string generated by this predicate contains the first character of every word. Conditional trigrams are formed from the first characters of three consecutive words[9].

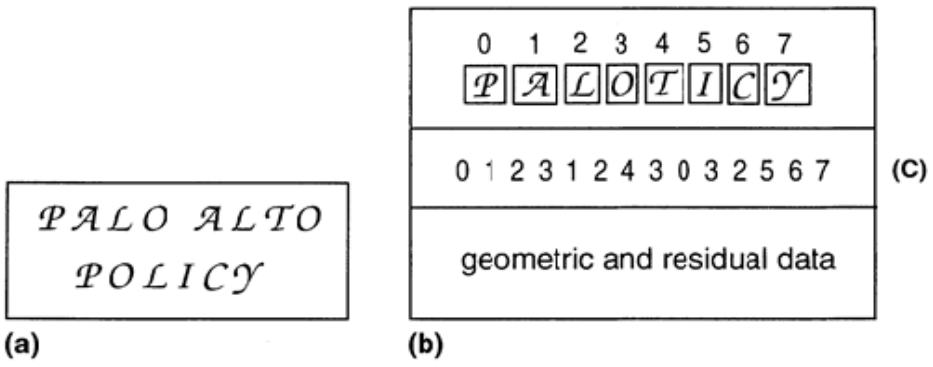

Figure 4: Depiction of symbolic compression, showing an original image (a) and the compressed image (b). The sequence of identifier's in (c) encodes the order of characters in the original document.

V. DOCUMENT DECIPHERING

To implement proposed approach the HMM model are used. In an even more realistic scenario, a single pattern could correspond to a partial symbol or multiple symbols due to image fragmentation and segmentation errors. A deciphering algorithm is available for simple substitution ciphers. By exploiting the redundancy in a language, the plain text message can be recovered from a sequence of cipher symbols of sufficient length. Numerous algorithmic solutions have been proposed for simple

substitution ciphers, including relaxation techniques dictionary-based pattern matching and optimization techniques. We propose a deciphering algorithm that uses a Hidden Markov Model (HMM) . Considering the Markov process of state traversal as a language source from which a particular plain text message can be generated with some probability, then the added symbol production at the traversed states in an HMM describes the enciphering process of a substitution cipher, where each letter in plain text is replaced with a cipher symbol one at a time.

This analogy between the source language modeling as a Markov process and the representation of the enciphering function by symbol probabilities is the basis for our solution. The state probabilities are initialized with language statistics, and the symbol probabilities are estimated with the EM algorithm. Information extraction from symbolically compressed documents can be viewed as a deciphering problem. The objective is the recovery of the association between character interpretations and pattern templates from a sequence of template identifiers. In symbolic compression schemes, image components are grouped to improve clustering and they are also roughly sorted in reading order to the reduce entropy in their relative offsets.

The objective of both measures is to improve compression performance. However, they also facilitate the application of deciphering techniques for information extraction. Figure shows an outline of the proposed HMM deciphering algorithm. It reads the pattern identifier sequence from a symbolically compressed document image and uses character transition probabilities to produce partial OCR results.

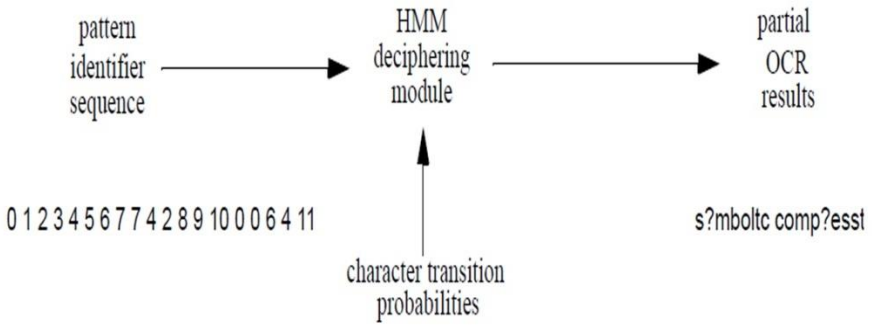

Figure 5: Deciphering a symbolic compressed image produces partial OCR results.

These results may not be completely correct. However, they are often adequate for various tasks which will be described in the next section First of all, it is obvious that the problem is never truly a simple substitution. Even with ample exemplars, certain contents such as numeric strings cannot be deciphered due to lack of context. Nevertheless, we believe sufficient information can be recovered for language identification, duplicate detection or document classification. Identification of the language of the text in the original image can also be performed by a version of the HMM deciphering algorithm.

- **Deciphering by Hidden Markov Models**

Numerous solutions for the deciphering problem have been proposed, including relaxation algorithms [13][7], dictionary based pattern matching [12], and optimization techniques [3][16]. We propose a solution that uses the well-developed theory of Hidden Markov Models [15]. The abstraction of state transitions and observable symbols in an HMM is analogous to the separation of a Markov language source and subsequent enciphering step.

Markov models have been used for natural language modeling. If we accept the Markov process of state traversal as a language source from which a particular plain text message can be generated with some probability, then the added symbol production at the traversed states in a hidden Markov model perfectly describes the enciphering procedure of a monographic substitution cipher, where each letter in plain text is replaced with a cipher symbol one at a time. This analogy between source language modeling as a Markov process and representation of the enciphering function by symbol probabilities is the basis for our solution, as shown in Figure.

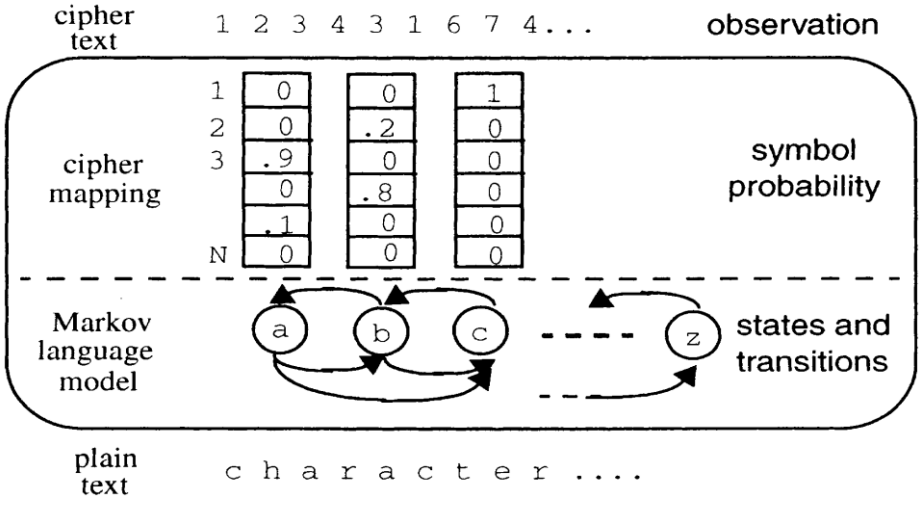

Figure 6: In the hidden Marko approach, the deciphering problem is formulated as ending the enciphering mapping that most likely produced the observed cipher text for the underlying Markov language source.

In a first order model, there are n states, each representing a letter in the plain text alphabet. Associated with each state, is a state transition probability function, and a symbol probability function. The first state in a sequence is selected according to an initial probability. Subsequent states are generated according to the transition probabilities, outputting one of the cipher symbols at each state with probability. Figure 5 - In the hidden Markov approach, the deciphering problem is formulated as finding the enciphering mapping that most likely produced the observed cipher text for the underlying Markov language source. The initial state probability is simply the character frequency of . Both the initial and transition probabilities are estimated from a corpus of the source language and remain fixed, providing a first order Markov modeling of the source language. Symbol probabilities are estimated using the forward-backward algorithm [15]. The initial estimation is defined as where is calculated from a cipher of length with occurrences of symbol using a binomial distribution. To determine the decipher mapping, we assign a plain symbol that most likely corresponds to each cipher symbol. Since is conditioned on the cipher symbol, the following decision criterion is used.

It should pointed out that we have explicitly constructed the deciphering function from the estimated symbol probabilities. However, it is unnecessary, even circuitous, to produce the underlying plain text this way. With fixed transition probabilities and estimated symbol probabilities, the Viterbi algorithm can be used to find the most likely sequence of states through which the observed symbols are produced. This state sequence, corresponding to the most likely plain text from which the cipher text is generated, given our parameter estimations, can be used directly for evaluation. Contrary to the deciphering solution where all occurrences of a cipher symbol in the cipher text must decode into the same letter in plain text, the most probable plain text generated from a state sequence may not have a consistent one-to-one mapping to the cipher text. Constructing the deciphering function incorporates the likelihood of all possible paths and has shown better results in our experiments than the direct method.

VI. RESULTS AND DISCUSSION

From the above discussion it is observed that duplicate detection on compressed images is better performance then the uncompressed images so we described the different methods for performing document duplicate detection directly on images in a symbolic compression format. Since the language statistics inherent in document content are largely preserved in the sequence of cluster identifiers, the original character interpretations can be recovered with a deciphering algorithm. We proposed an HMM solution for the deciphering problem. While the overall character interpretation rates are not perfect, we demonstrated that sufficient information is recovered for document duplicate detection. This offers an efficient and versatile solution to detecting full and partial duplicates. It also provides a useful method for indexing large document databases. Future work will consider implementation of this technique in the IM3 system.

VII. CONCLUSION AND FUTURE SCOPE

Duplicate document image is the big problem, a method was presented for performing document duplicate detection directly on images. Since, the language statistics inherent in document content are largely preserved in the sequence of cluster identifiers, the original character interpretations can be recovered with a deciphering algorithm. We proposed a hash function solution with HMM for deciphering problem. While the overall character interpretation rates are 90% perfect, we demonstrate that sufficient information is recovered for document duplicate detection. This offers an efficient and versatile solution to detecting full and partial duplicates.

The future of image processing will involve scanning the heavens for other intelligent life out in space. Also new intelligent, digital species created entirely by research scientists in various nations of the world will include advances in image processing applications. Due to advances in image processing and related technologies there will be millions and millions of robots in the world in a few decades time, transforming the way the world is managed. With increasing power and sophistication of modern computing, the concept of computation can go beyond the present limits and in future, image processing technology will advance and the visual system of man can be replicated. The future trend in remote sensing will be towards improved sensors that record the same scene in many spectral channels. Graphics data is becoming increasingly important in image processing app1ications. The future image processing applications of satellite based imaging ranges from planetary exploration to surveillance applications.

In future, the complete implementation of the system will be consider and implement on IM3 server. IM3 helps to collaborative and highly impressive results.

REFERENCES

[1] Ascher, R.N., Nagy, G., 1974. A means for achieving a high degree of compaction on scan-digitized printed text. IEEE Trans. Comput. C-23 (11), 1174±1179.

[2] Casey, R., Nagy, G., 1968. Autonomous reading machine.IEEE Trans. Comput. C-7.

[3] Howard, P., Kossentini, F., Martins, B., Forchhammer, S., Rucklidge, W.J., 1998.The emerging JBIG2 standard. IEEE Trans. Circuits Systems Video Technol. 8 (7), 838±848.

[4] Hull, J.J., Hart, P., 1998.The infinite memory multifunction machine. In: Pre-proceedings 3rd IAPR Workshop on Document Analysis Systems, Nagano, Japan, 4±6 Novem- ber, pp. 49±58.

[5] Hull, J.J., Lee, D.-S., Cullen, J., Hart, P., 1999.Document analysis techniques for the infinite memory multifunction machine. In: Proc. 10th Internat. Workshop on Database and Expert System Applications, Florence, Italy, 1±3 September, pp. 561±565.

[6] King, J., Bahler, D., 1992. An implementation of probabilistic relaxation in the cryptanalysis of simple substitution ciphers. Cryptologia 16 (3), 215±225.

[7] Lee, D.-S., Hull, J.J., 1999.Information extraction from symbolically compressed document images. In: Proc. 1999 Symposium on Document Image Understanding Technology, Annapolis, MD, 14±16 April, pp. 176±182.

[8] Lee, D.-S., Hull, J.J., 1999. Duplicate detection for symbolically compressed documents. In: Proc. 5th Internat. Conf.Document Analysis and Recognition, Bangalore, India, 20±22 September, pp. 305±308.

[9] Peleg, S., Rosenfeld, A., 1979. Breaking substitution ciphers using a relaxation algorithm. Commun. ACM 22 (11), 598± 605.

[10] Phillips, I.T., Chen, S., Haralick, R.M., 1993. CD-ROM document database standard. In: Proc. 2nd ICDAR, pp. 478±483.

[11] Rabiner, L.R., Juang, B.H., 1986. An introduction to hidden Markov models. IEEE ASSP Magazine, 4±16.

[12] Witten, I., Moffat, A., Bell, T., 1994. Managing Gigabytes: Compressing and Indexing Documents and Images. Van Nostrand Reinhold, New York.

[13] K. Mohiuddin, J. Rissanen and R. Arps, "Lossless binary image compression based on pattern matching," Proceedings of International Conference on Computers, Systems & Signal Processing, December, 1984.

[14] G. Nagy, S. Seth and K. Einspahr, "Decoding substitution ciphers by means of word matching with application to OCR," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-9, no. 5, pp. 710-715, 1987.

[15] S. Peleg and A. Rosenfeld, "Breaking substitution ciphers using a relaxation algorithm," Communications of the ACM, vol.22, no.11, pp. 598-605, November 1979.

[16] I. T. Phillips, S. Chen, R. M. Haralick, "CD-ROM document database standard," Proceedings of the 2nd ICDAR, pp. 478-483, 1993.

[17] L. R. Rabiner and B. H. Juang, "An introduction to hidden markov models," IEEE ASSP Magazine, pp. 4-16, January 1986.

[18] R. Spillman, M. Janssen, B. Nelson and M. Kepner, "Use of a genetic algorithm in the cryptanalysis of simple substitution ciphers," Cryptologia, vol. 17, no. 1, pp. 31-44, 1993.

[19] I. Witten, T. Bell, H. Emberson, S. Inglis, and A. Moffat, "Textual Image Compression: two stage lossy/lossless encoding of textual images," Proceedings of the IEEE, vol. 82, no. 6, pp. 878-888, June 1994.

[20] I. Witten, A. Moffat and T. Bell, "Managing Gigabytes: Compressing and Indexing Documents and Images," Van Nostrand Reinhold, New York, 1994.