**1 Introduction**

The Indian Postal System (IPS) is one among the few public sector units which deserve to be fully automated. The time taken for a mail delivery can be viewed as the sum of the transit time and processing time. The transit time is something which is inevitable. But, the processing time, which includes the time for mail address interpretation and sorting, can be reduced to a great deal if the sorting process is automated. Consider a mail that is posted from Coimbatore to Shimla. The existing postal system involves human intervention in the processing of this mail at least four times: Coimbatore, Chennai, Delhi and Shimla. At each of these points, a human reads the mail and manually interprets it before sorting it according to its destination address. This process of mail sorting is highly time consuming besides being error prone.

In this project we intend to propose an Automatic Mail Processor (AMP) which scans and interprets the destination address and converts it into a Delivery Point Code, which is printed on the mail in the form of a bar-code. Now that the destination address is in the form of a barcode, it can be interpreted by using a low-cost barcode reader for future sorting processes.

* 1. **Implementation Consideration**

The Automatic Mail Processor (AMP) is designed to be compatible with all kinds of mails such as Inland letters, Post Cards, Post Covers, and Envelopes etc. The job of AMP is simplified in the first three cases due to the presence of predefined address locations. In the case of an Envelope, there is an additional work of locating the position of the destination address.

Let us consider such an Envelope with NO predefined grids as an example. All the stages in this project are described with Envelope as an instance. The same can be implemented with much ease for all other types of mails due to their inherent simplicities.

It should also be ensured that sufficient space is provided in each mail for printing the bar-code. Though space would not be a constraint in large Envelopes, sufficient care has to be taken in the case of small Post Cards and Covers.

1. **Automatic Mail Processor**

In this chapter we discuss the necessary step of our Automatic Mail Processor system.

[1] In this overview, character recognition (CR) is used as an umbrella term, which covers all types of machine recognition of characters in various application domains. The overview serves as an update for the state-of-the-art in the CR field, emphasizing the methodologies required for the increasing needs in newly emerging areas, such as development of electronic libraries, multimedia databases, and systems which require handwriting data entry. The study investigates the direction of the CR research, analysing the limitations of methodologies for the systems, which can be classified based upon two major criteria:

1) The data acquisition process (on-line or off-line) and

2) The text type (machine-printed or handwritten).

No matter in which class the problem belongs, in general, there are five major stages in the CR problem:

1) Pre-processing;

2) Segmentation;

3) Representation;

4) Training and Recognition;

5) Post-Processing.

Although the off-line and on-line CR techniques have different approaches, they share a lot of common problems and solutions.

The report also reviews some of the methodologies for on-line CR, as it intersects with the off-line case. After giving a historical review of the developments in Chapter 3, the block diagram of the entire system is represented in Chapter 4, literature review of the methodologies of CR systems are reviewed in Chapter 5. Implementation of Automatic Mail Processor is discussed in Chapter 7. Finally, future research directions are discussed in Chapter 11.s

1. **History**

[1] Writing, which has been the most natural mode of collecting, storing, and transmitting information through the centuries, now serves not only for communication among humans but also serves for communication of humans and machines. The intensive research effort in the field of CR was not only because of its challenge on simulation of human reading but also because it provides efficient applications such as the automatic processing of bulk amount of papers, transferring data into machines, and web interface to paper documents. Historically, CR systems have evolved in three ages.

1900–1980 Early Ages: The history of CR can be traced as early as 1900, when the Russian scientist Tyuring attempted to develop an aid for the visually handicapped. The first character recognizers appeared in the middle of the 1940s with the development of digital computers. The early work on the automatic recognition of characters has been concentrated either upon machine-printed text or upon a small set of well-distinguished handwritten text or symbols. Machine-printed CR systems in this period generally used template matching in which an image is compared to a library of images. For handwritten text, low-level image processing techniques have been used on the binary image to extract feature vectors, which are then fed to statistical classifiers. Successful, but constrained algorithms have been implemented mostly for Latin characters and numerals. However, some studies on Japanese, Chinese, Hebrew, Indian, Cyrillic, Greek, and Arabic characters and numerals in both machine-printed and handwritten cases were also initiated.

The commercial character recognizers were available in the 1950s, when electronic tablets capturing the x–y coordinate data of pen-tip movement was first introduced. This innovation enabled the researchers to work on the on-line handwriting recognition problem. A good source of references for on-line recognition until 1980.

1980–1990 Developments: Studies up until 1980 suffered from the lack of powerful computer hardware and data acquisition devices. With the explosion of information technology, the previously developed methodologies found a very fertile environment for rapid growth in many application areas, as well as CR system development. Structural approaches were initiated in many systems in addition to the statistical methods. The CR research was focused basically on the shape recognition techniques without using any semantic information. This led to an upper limit in the recognition rate, which was not sufficient in many practical applications.

After 1990 Advancements: The real progress on CR systems is achieved during this period, using the new development tools and -methodologies, which are empowered by the continuously growing information technologies.

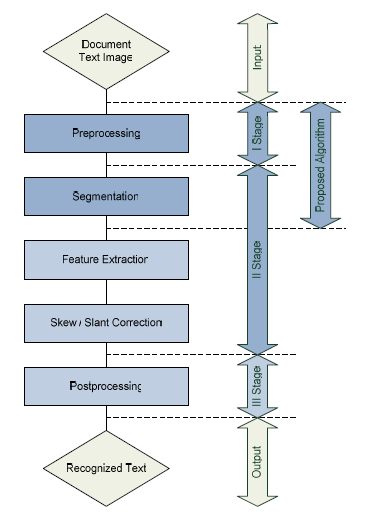
In the early 1990s, image processing and pattern recognition techniques were efficiently combined with artificial intelligence (AI) methodologies. Researchers developed complex CR algorithms, which receive high-resolution input data and require extensive number crunching in the implementation phase. Nowadays, in addition to the more powerful computers and more accurate electronic equipment’s such as scanners, cameras, and electronic tablets, we have efficient, modern use of methodologies such as neural networks (NNs), hidden Markov models (HMMs), fuzzy set reasoning, and natural language processing. The recent systems for the machine-printed off-line and limited vocabulary, user-dependent on-line handwritten characters are quite satisfactory for restricted applications. However, there is still a long way to go in order to reach the ultimate goal of machine simulation of fluent human reading, especially for unconstrained on-line and off-line handwriting.

1. **Block Diagram**

Although document conversion system incorporates scanning, binarization, region segmentation, text recognition and document analysis, its procedure can be divided into three main stages: pre-processing, processing and post-processing, such as shown in Figure 1.

In the first stage, algorithms for document text image binarization and normalization are applied. After pre-processing stage, text is prepared for segmentation, feature extraction and character recognition. During the second stage, algorithm for text segmentation, skew rate and reference text line identification is enforced. Also, reference text based on skew and stroke angle, is straightened and repaired. Finally, in third stage character recognition process is applied.

The Block Diagram of AMP begins with a Pre-Processing unit which comprises of two stages namely Digitization and Binarization. Denoising can be considered to be a part of the Digitization stage. The Address Block Location module determines the location of destination address in the mail as show in figure 2. The Line and Word separation units divide the extracted address into constituent lines and words by employing Horizontal and Vertical Scanning algorithms, as shown in figure 3. The Address Parsing and Recognition module parses and recognizes the imperative fields of the destination address. From the recognized fields, the Delivery Point Code (DPC) is generated by the DPC unit. The DPC is subsequently converted into a barcode that will be printed on the mail, as shown in figure 4.



*Figure 1: Block Diagram [2]*

*Figure 2: Pre Processing Steps.*

*Figure 3: Segmentation Hierarchy.*

*Figure 4: Post Processing Steps.*

1. **Literature Review: Methodologies of CR Systems**

In this section, we focus on the methodologies of CR systems, emphasizing the off-line handwriting recognition problem. A bottom–up approach for most of the systems would be starting the process from the pixel level and ending up with a meaningful text. This approach varies a great deal, depending upon the type of CR system and the methodology used. The literature review in the field of CR indicates that these hierarchical tasks are grouped in the stages of the CR for pre-processing, segmentation, representation, training and recognition, and post-processing. In some methods, some of the stages are merged or omitted; in others a feedback mechanism is used to update the output of each stage.

* 1. **Preprocessing**

The raw data, depending on the data acquisition type, is subjected to a number of preliminary processing steps to make it usable in the descriptive stages of character analysis. Preprocessing aims to produce data that are easy for the CR systems to operate accurately.

The main objectives of preprocessing are

1) Noise reduction;

2) Normalization of the data;

3) Compression in the amount of information to be retained.

In order to achieve the above objectives, the following techniques are used in the preprocessing stage.

* + 1. **Noise Reduction:**

The noise, introduced by the optical scanning device or the writing instrument, causes disconnected line segments, bumps and gaps in lines, filled loops, etc. The distortion, including local variations, rounding of corners, dilation, and erosion, is also a problem. Prior to the CR, it is necessary to eliminate these imperfections. Hundreds of available noise reduction techniques can be categorized in three major groups [1], [3].

* + - 1. **Filtering:**

This aims to remove noise and diminish spurious points, usually introduced by uneven writing surface and/or poor sampling rate of the data acquisition device. Various spatial and frequency domain filters can be designed for this purpose. The basic idea is to convolute a predefined mask with the image to assign a value to a pixel as a function of the grey values of its neighbouring pixels. Filters can be designed for smoothing [4], sharpening [1], thresholding [5], removing slightly textured or colored background [6], and contrast adjustment purposes [7].

* + - 1. **Morphological Operations:**

The basic idea behind the morphological operations is to filter the document image replacing the convolution operation by the logical operations. Various morphological operations can be designed to connect the broken strokes, decompose the connected strokes, smooth the contours, prune the wild points [8], thin the characters [9], and extract the boundaries [10]. Therefore, morphological operations can be successfully used to remove the noise on the document images due to low quality of paper and ink, as well as erratic hand movement.

* + - 1. **Noise Modelling:**

Noise could be removed by some calibration techniques if a model for it were available. However, modelling the noise is not possible in most of the applications. There is very little work on modelling the noise introduced by optical distortion, such as speckle, skew, and blur. Nevertheless, it is possible to assess the quality of the documents and remove the noise to a certain degree, as suggested in [1].

* + 1. **Normalization:**

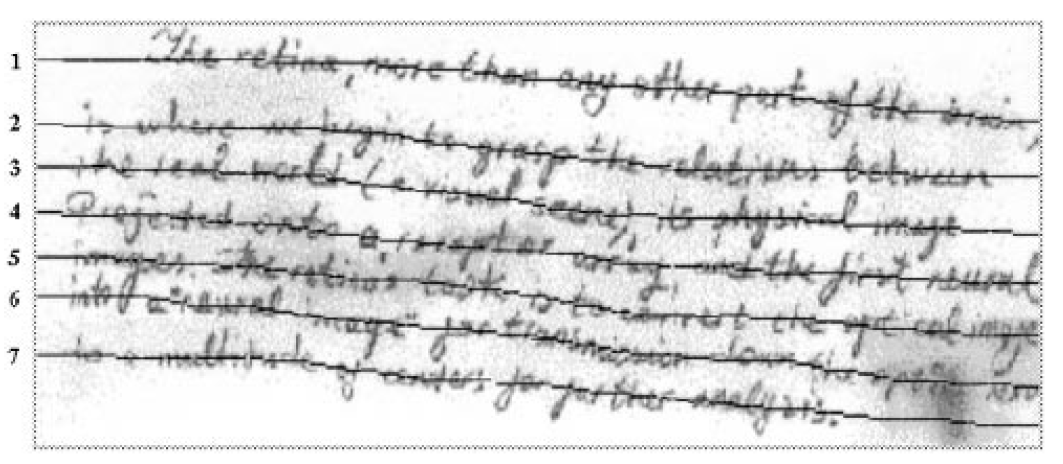
Normalization methods aim to remove the variations of the writing and obtain standardized data. The following are the basic methods for normalization [1].

* + - 1. **Skew Normalization and Baseline Extraction:**

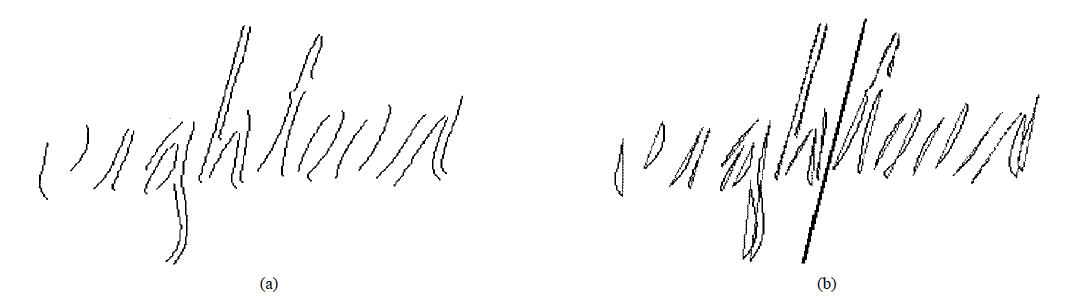
Due to inaccuracies in the scanning process and writing style, the writing may be slightly tilted or curved within the image. This can hurt the effectiveness of later algorithms and, therefore, should be detected and corrected. Additionally, some characters are distinguished according to the relative position with respect to the baseline (e.g., “9” and “g”). Methods of baseline extraction include using the projection profile of the image [11], a form of nearest neighbours clustering [12], cross correlation method between lines [13], and using the Hough transform [14]. In [15], an attractive repulsive NN is used for extracting the baseline of complicated handwriting in heavy noise (see figure 5). After skew detection, the character or word is translated to the origin, rotated, or stretched until the baseline is horizontal and retranslated back into the display screen space.

* + - 1. **Slant Normalization:**

One of the measurable factors of different handwriting styles is the slant angle between longest stroke in a word and the vertical direction. Slant normalization is used to normalize all characters to a standard form. The most common method for slant estimation is the calculation of the average angle of near-vertical elements (see Figure 6). The vertical line elements from contours are extracted by tracing chain code components using a pair of one-dimensional (1-D) filters. Coordinates of the start and end points of each line element provide the slant angle. Another study [1] uses an approach in which projection profiles are computed for a number of angles away from the vertical direction. The angle corresponding to the projection with the greatest positive derivative is used to detect the least amount of overlap between vertical strokes and, therefore, the dominant slant angle. In [1], slant detection is performed by dividing the image into vertical and horizontal windows. The slant is estimated based on the center of gravity of the upper and lower half of each window averaged over all the windows. Finally, in [1], a variant of the Hough transform is used by scanning left to right across the image and calculating projections in the direction of 21 different slants. The top three projections for any slant are added and the slant with the largest count is taken as the slant value. On the other hand, in some studies, recognition systems do not use slant correction and compensate it during training stage [1].



*Figure 5: Baseline extraction using attractive and repulsive network. [1]*



*Figure 6: Slant angle estimation. (a) Near vertical elements. (b) Average slant angle. [1]*

* + - 1. **Size Normalization:**

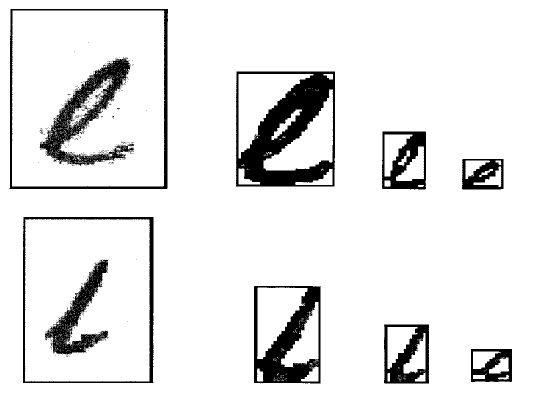
This is used to adjust the character size to a certain standard. Methods of CR may apply both horizontal and vertical size normalizations. The character is divided into number of zones and each of these zones is separately scaled. Size normalization can also be performed as a part of the training stage, and the size parameters are estimated separately for each particular training data. In Figure 7, two sample characters are gradually shrunk to the optimal size, which maximize the recognition rate in the training data. On the other hand, word recognition, due to the desire to preserve large intraclass differences in the length of words so they may assist in recognition, tends to only involve vertical height normalization or bases the horizontal size normalization on the scale factor calculated for the vertical normalization [1].

* + - 1. **Contour Smoothing:**

It eliminates the errors due to the erratic hand motion during the writing. It generally reduces the number of sample points needed to represent the script, thus improving efficiency in remaining preprocessing steps [4].

* + 1. **Compression:**

It is well known that classical image compression techniques transform the image from the space domain to domains, which are not suitable for recognition. Compression for CR requires space domain techniques for preserving the shape information. Two popular compression techniques are thresholding and thinning.



*Figure 7:* Normalization of characters “e” and “l”.  *[1]*

* + - 1. **Thresholding:**

In order to reduce storage requirements and to increase processing speed, it is often desirable to represent gray-scale or color images as binary images by picking a threshold value. Two categories of thresholding exist: global and local. Global thresholding picks one threshold value for the entire document image which is often based on an estimation of the background level from the intensity histogram of the image. Local (adaptive) thresholding use different values for each pixel according to the local area information. A comparison of common global and local thresholding techniques is given by using an evaluation criterion that is goal-directed in the sense that the accuracies of a CR system using different techniques were compared. On those tested, it is shown that Niblack’s locally adaptive method [1] produces the best result.

* + - 1. **Thinning:**

While it provides a tremendous reduction in data size, thinning extracts the shape information of the characters. Thinning can be considered as conversion of off-line handwriting to almost on-line like data, with spurious branches and artifacts. Two basic approaches for thinning are 1) pixel wise and 2) nonpixel wise thinning. Pixel wise thinning methods locally and iteratively process the image until one pixel wide skeleton remains. They are very sensitive to noise and may deform the shape of the character. On the other hand, the nonpixel wise methods use some global information about the character during the thinning. They produce a certain median or centerline of the pattern directly without examining all the individual pixels. In [1], clustering-based thinning method defines the skeleton of character as the cluster centres. Some thinning algorithms identify the singular points of the characters, such as end points, cross points, and loops. These points are the source of problems. In a nonpixel wise thinning, they are handled with global approaches. The iterations for thinning can be performed either in sequential or parallel algorithms. Sequential algorithms examine the contour points by raster scan or contour following. Parallel algorithms are superior to sequential ones since they examine all the pixels simultaneously, using the same set of conditions for deletion. They can be efficiently implemented in parallel hardware. The preprocessing techniques are well explored and applied in many areas of image processing besides CR. [1] Note that the above techniques affect the data and may introduce unexpected distortions to the document image. As a result, these techniques may cause the loss of important information about writing. They should be applied with care.

* 1. **Segmentation**

The preprocessing stage yields a “clean” document in the sense that a sufficient amount of shape information, high compression, and low noise on a normalized image is obtained. The next stage is segmenting the document into its subcomponents. Segmentation is an important stage because the extent one can reach in separation of words, lines, or characters directly affects the recognition rate of the script. There are two types of segmentation: external segmentation, which is the isolation of various writing units, such as paragraphs, sentences, or words, and internal segmentation, which is the isolation of letters, especially in cursively written words.

* + 1. **External Segmentation**

It is the most critical part of the document analysis, which is a necessary step prior to the off-line CR Although document analysis is a relatively different research area with its own methodologies and techniques, segmenting the document image into text and non-text regions is an integral part of the OCR software. Therefore, one who works in the CR field should have a general overview for document analysis techniques.

Page layout analysis is accomplished in two stages: The first stage is the structural analysis, which is concerned with the segmentation of the image into blocks of document components (paragraph, row, word, etc.), and the second one is the functional analysis, which uses location, size, and various layout rules to label the functional content of document components (title, abstract, etc.).

A number of approaches regard a homogeneous region in a document image as a textured region. Page segmentation is then implemented by finding textured regions in gray-scale or color images. For example, Jain et al. use Gabor filtering and mask convolution [1], the Tang et al. approach is based on fractal signature [16], and Doermann’s method [1] employs wavelet multiscale analysis. Many approaches for page segmentation concentrate on processing background pixels or using the white space in a page to identify homogeneous regions [1]. These techniques include X–Y tree [17], pixel-based projection profile [18], connected component-based projection profile [1], white space tracing [19], and white space thinning [1].

They can be regarded as top–down approaches, which segment a page, recursively, by X-cut and Y-cut from large components, starting with the whole page to small components, eventually reaching individual characters. On the other hand, there are some bottom–up methods which recursively grow the homogeneous regions from small components based on the processing on pixels and connected components. An example of this approach may be the Docstrum method, which uses, nearest neighbour clustering. Some techniques combine both top–down and bottom–up techniques.

* + 1. **Internal Segmentation:**

Although the methods have developed remarkably in the last decade and a variety of techniques have emerged, segmentation of cursive script into letters is still an unsolved problem.

Character segmentation strategies are divided into three categories [1].

* + - 1. **Explicit Segmentation:**

In this strategy, the segments are identified based on “character-like” properties. The process of cutting up the image into meaningful components is given a special name: dissection.

Dissection is a process that analyses an image without using a specific class of shape information. The criterion for good segmentation is the agreement of general properties of the segments with those expected for valid characters.

Available methods based on the dissection of an image use white space and pitch, vertical projection analysis [20], connected component analysis [1], and landmarks [21].

* + - 1. **Implicit Segmentation:**

This segmentation strategy is based on recognition. It searches the image for components that match predefined classes. Segmentation is performed by the use of recognition confidence, including syntactic or semantic correctness of the overall result.

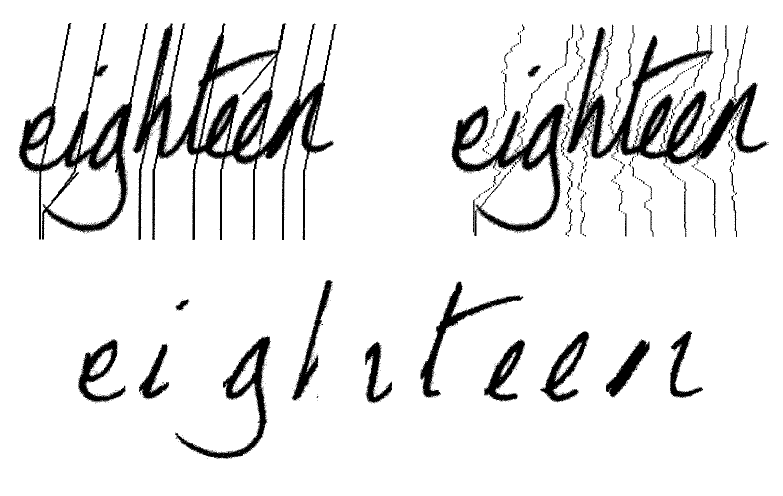
In this approach, two classes of methods can be employed:

1) Methods that make some search process and

2) Methods that segment a feature representation of the image.

The first class attempts to segment words into letters or other units without use of feature-based dissection algorithms. Rather, the image is divided systematically into many overlapping pieces without regard to content. Conceptually, these methods originate from schemes developed for the recognition of machine-printed words [22]. The basic principle is to use a mobile window of variable width to provide sequences of tentative segmentations, which are confirmed by CR. Another technique combines dynamic programming and NNs. Finally, the method of selective attention takes NNs even further in the handling of the segmentation problem [23].

The second class of methods segments the image implicitly by classification of subsets of spatial features collected from the image as a whole. This approach can be divided into two categories: HMM-based approaches and non-Markov-based approaches. The survey in [24] provides an introduction to HMM-based approaches in recognition applications. HMMs are used to structure the entire word recognition process. Non-Markov approaches stem from concepts used in machine vision for recognition of occluded object [25]. This family of recognition-based approaches uses probabilistic relaxation, the concept of regularities and singularities, and backward matching.



(c)

(b)

(a)

*Figure 8: Segmentation by finding the shortest path of a graph formed by gray level image. (a) Segmentation intervals. (b) Segmentation paths. (c) Segments.  [1]*

* + - 1. **Mixed Strategies:**

They combine explicit and implicit segmentation in a hybrid way. A dissection algorithm is applied to the image, but the intent is to “over segment,” i.e., to cut the image in sufficiently many places that the correct segmentation boundaries are included among the cuts made. Once this is assured, the optimal segmentation is sought by evaluation of subsets of the cuts made.

Each subset implies a segmentation hypothesis, and classification is brought to bear to evaluate the different hypothesis and choose the most promising segmentation. In [26], the segmentation problem is formulated as finding the shortest path of a graph formed by binary and gray-level document image. The HMM probabilities, obtained from the characters of a dissection algorithm, are used to form a graph. The optimum path of this graph improves the result of the segmentation by dissection and HMM recognition. Figure 8 (a) indicates the initial segmentation intervals obtained by evaluating the local maxima and minima together with the slant angle information. Figure 8 (b) and (c) show the shortest path for each segmentation interval and the resulting candidate characters, respectively. Mixed strategies yield better results compared to explicit and implicit segmentation methods.

The techniques presented above have limited capabilities in segmentation. Error detection and correction mechanisms should be embedded into the systems for which they were developed. Implementing such mechanisms results in superior performance, and efficient system.

**5.3 Representation**

Image representation plays one of the most important roles in a recognition system. In the simplest case, gray-level or binary images are fed to a recognizer. However, in most of the recognition systems, in order to avoid extra complexity and to increase the accuracy of the algorithms, a more compact and characteristic representation is required. For this purpose, a set of features is extracted for each class that helps distinguish it from other classes while remaining invariant to characteristic differences within the class.

In the following, hundreds of document image representation methods are categorized into three major groups.

**5.3.1 Global Transformation and Series Expansion**

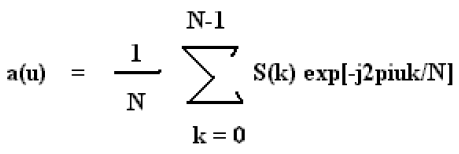
A continuous signal generally contains more information than needs to be represented for the purpose of classification. This may be true for discrete approximations of continuous signals as well. One way to represent a signal is by a linear combination of a series of simpler well-defined functions. The coefficients of the linear combination provide a compact encoding known as transformation or/and series expansion. Deformations like translation and rotations are invariant under global transformation and series expansion. Common transform and series expansion methods used in the CR field are include the following.

**5.3.1.1 Fourier Transforms**

The general procedure is to choose magnitude spectrum of the measurement vector as the features in an n-dimensional Euclidean space. One of the most attractive properties of the Fourier transform is the ability to recognize the position-shifted characters, when it observes the magnitude spectrum and ignores the phase. Fourier transforms have been applied to CR in many ways [1].

The Fourier descriptors (FDs) of the character are computed by applying DFT on the chain code points. DFT is applied in such a way that the x-coordinate x(k) of the chain code is considered as the real part and the y-coordinate y(k) is considered as the imaginary part.

S(k) = x(k) + jy(k)



The complex coefficients a(u) for u = 0, 1, 2, 3, 4, 5,… are called the Fourier descriptors of the boundary. The number of DFT components obtained depends on the size of the contour (i.e.) the number of pixels in the contour.

For identifying a character, all the DFT components are not required. A satisfactory description can be obtained by choosing few lower frequency and few higher frequency components. Larger the number of components considered, higher is the accuracy. But, as the number of components considered increases, computational complexity is observed. So, a trade-off has to be made between accuracy and complexity while choosing the number of DFT components considered for recognition. In this project, 10 low frequency components and 10 high frequency components are considered for each character. To summarize, every character will be represented by a set of 20 DFT components.

**5.3.1.2 Gabor Transform**

This is a variation of the windowed Fourier transform. In this case, the window used is not a discrete size but is defined by a Gaussian function [27].

**5.3.1.3 Wavelets**

Wavelet transformation is a series expansion technique that allows us to represent the signal at different levels of resolution. The segments of document image, which may correspond to letters or words, are represented by wavelet coefficients, corresponding to various levels of resolution. These coefficients are then fed to a classifier for recognition [28]. The representation in multi resolution analysis (MRA) with low resolution can absorb the local variation in handwriting as opposed to MRA with high resolution. However, the representation in low resolution may cause the important details for the recognition stage to be lost.

**5.3.1.4 Moments:**

Moments, such as central moments, Legendre moments, and Zernike moments, form a compact representation of the original document image that make the process of recognizing an object scale, translation, and rotation invariant [1]. Moments are considered as series expansion representation since the original image can be completely reconstructed from the moment coefficients.

**5.3.1.5 Karhunen–Loeve Expansion:**

This is an eigenvector analysis which attempts to reduce the dimension of the feature set by creating new features that are linear combinations of the original ones. It is the only optimal transform in terms of information compression. Karhunen–Loeve expansion is used in several pattern recognition problems such as face recognition. It is also used in the National Institute of Standards and Technology (NIST) OCR system for form-based handprint recognition [1]. Since it requires computationally complex algorithms, the use of Karhunen–Loeve features in CR problems is not widespread.

1. **Limitation of Prior Art**

In this study, we have overviewed the main approaches used in the CR field. Our attempt was to bring out the present status of CR research. Although each of the methods summarized above have their own superiorities and drawbacks, the presented recognition results of different methods seem very successful. Most of the recognition accuracy rates reported are over 85%. However, it is very difficult to make a judgment about the success of the results of recognition methods, especially in terms of recognition rates, because of different databases, constraints, and sample spaces. In spite of all the intensive research effort, numerous journal articles, conference proceedings, and patents, none of the proposed methods solve the CR problem out of the laboratory environment without putting constraints. The answer to the question “Where are we standing now?” is summarized in Table I.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | Machine Printed | | | Handwritten | | |
|  |  | Single Font | Omni Font | Multi Font | Discrete | Cursive | Mixed |
| On Line | Constrained |  |  |  |  |  |  |
| Unconstrained |  |  |  |  |  |  |
| Off Line | Noiseless |  |  |  |  |  |  |
| Noisy |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
|  | Well Done |  |
|  |  |  |
|  | Needs Improvement | |
|  |  |  |
|  | Needs More Research | |

*Table 1: Current Studies in CR Studies. [1]*

[1] A number of weaknesses, can be summarized as follows.

1) In all of the proposed methods, the studies on the stages of CR have come to a point where the improvements are marginal with the current research directions. The stages are mostly based on the shape extracting and recognition techniques and ignore the semantic information. Incorporation of the semantic information is not well explored. In most cases, it is too late to correct all the errors, which propagates through the stages of the CR, in the post processing stage. This situation implies the need of a global change in the approaches for freestyle handwriting.

2) A major difficulty lies behind the lack of the noise model, over all the stages. Therefore, many assumptions and parameters of the algorithms are set by trial and error at the initial phase. Unless there are strict constraints about the data and the application domain, the assumptions are not valid even for small changes out of the laboratory environment.

3) Handwriting generation involves semantic, syntactic, and lexical information, which is converted into a set of symbols to generate the pen-tip trajectory from a predefined alphabet. The available techniques suffer from the lack of characterizing the handwriting generation and the perceptual process in reading, which consists of many complicated phenomena. For example, none of the proposed systems take into account contextual anticipatory phenomena, which lead to co-articulations and the context effects on the writing [1]. The sequence of cursive letters is not produced in a serial manner, but parallel articulatory activity occurs. The production of a character is thus affected by the production of the surrounding characters and thus the context.

4) In most of the methods, recognition is isolated from training. The large amount of data is collected and used to train the classifier prior to the classification. Therefore, it is not easy to improve the recognition rate using the knowledge obtained from the analysis of the recognition errors.

5) Selection of the type and the number of features is done by heuristics. It is well known that design of the feature space depends on the training and recognition method. On the other hand, the performance of the recognizer highly depends on the selected features. This problem cannot be solved without evaluating the system with respect to the feature space and the recognition scheme. There exist no evaluation tools for measuring the performance of the stages as well as the overall performance of the system, indicating the source of errors for further improvements.

6) Although color scanners and tablets enable data acquisition with high resolution, there is always a trade-off between the data acquisition quality and complexity of the algorithms, which limits the recognition rate.

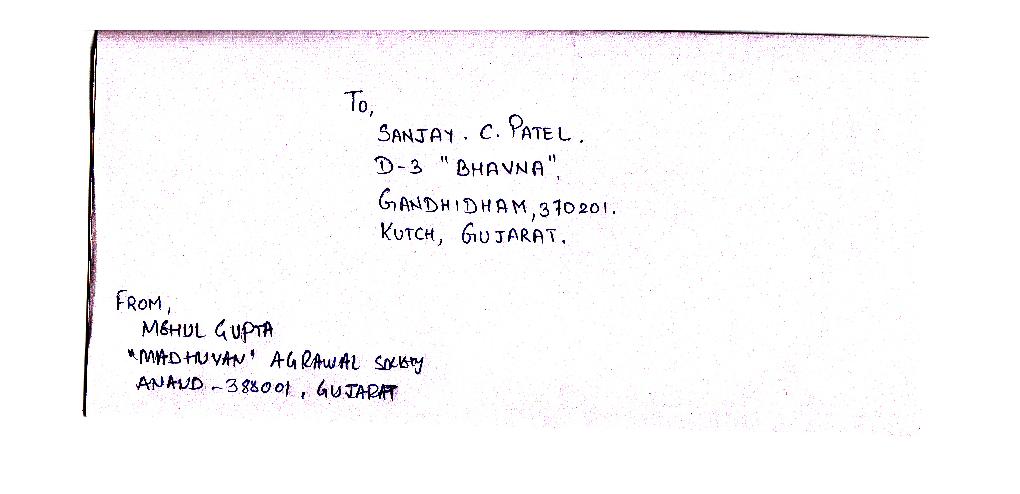
1. **Implementation of Automatic Mail Processor**

After considering the literature review, lets’ now look into the implementation considerations.

We have demonstrated the various steps required to recognise the characters from an image. The characters can be any of the two: Hand Written or Printed.

* 1. **Digitization**

The first step of AMP is to obtain a digitized image of the Envelope. Digitization is performed by scanning the Envelope with a high-resolution scanner to produce an 8-bit gray scaled image as shown below.

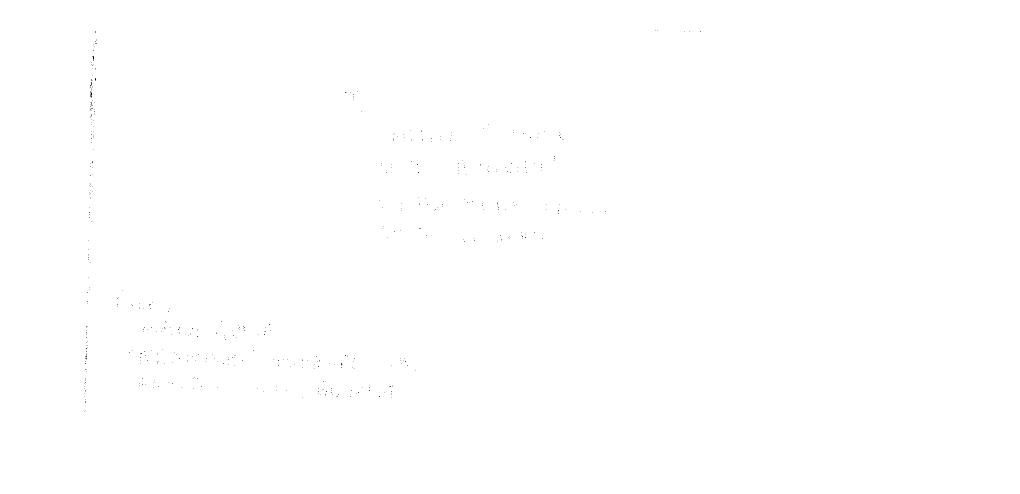
****

*Figure 9: Digitized Image*

* 1. **Noising Removal**

The digitized image usually contains certain noisy spurious pixels which can be removed by de-noising the image. The image de-noising is performed by subjecting the image to an averaging filter.

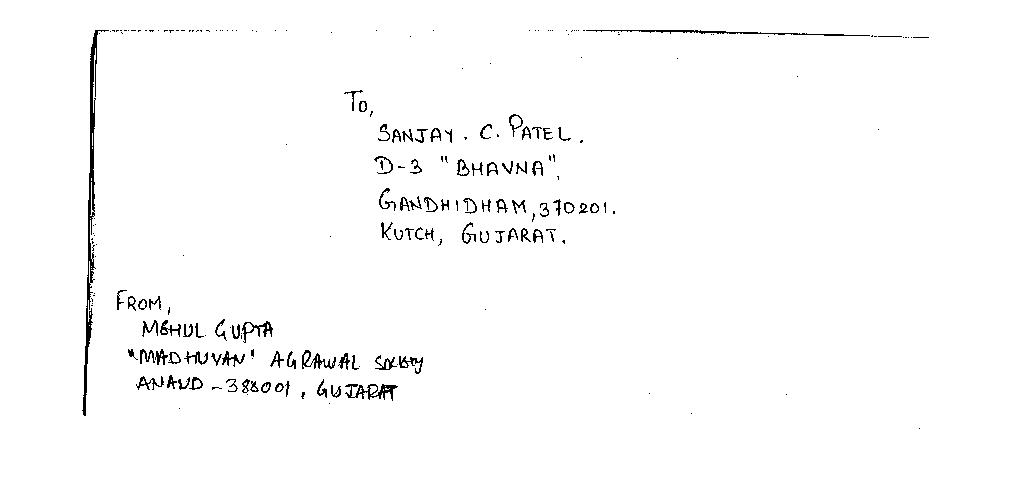
The mask is suitably set so as not to blur the image. A 3 X 3 mask would be sufficient to remove the discrete spurious noisy pixels from the scanned image of the mail. Median filtering may also be employed for this purpose.



*Figure 10: Noise Present in the Image*

* 1. **Binarization**

Every pixel of the averaged image is represented by 8-bits. But, for interpreting the destination address, a binary image with two colours is sufficient. Therefore, the averaged image is binarized by setting a threshold level, which is determined adaptively depending on the foreground and background colours. The binarized image is shown in Figure 11.



*Figure 11: De Noised and Binarized Image*

1. **Address Block Location**

The location of the destination address in an envelope is not pre-determined. This necessitates the need for an Address Block Location module (ABL). The ABL takes the binarized image from the preprocessor and determines the position of the destination address in the envelope. In other words, the ABL returns the coordinates of a rectangle with least area that can be drawn around the destination address after leaving a tolerance level. As the ABL distorts the original image while processing, a backup of the original image is stored for future retrieval.

The logic behind the working of ABL is the Redundancy Correction (RC) algorithm. The algorithm functions in such a way as to remove the explicit redundancies in the image such as the sender’s address and stamp impressions. The steps of this algorithm are illustrated below.

* 1. **Redundancy Correction (RC) Algorithm**

1. Obtain the image from pre processing unit

2. Low pass filter the image using a 20 X 20 spatial mask using point processing techniques.

3. As the low-pass filtered image is a non-binary image, re-binarize it by setting a high threshold value.

4. The Re-Binarization would cause the fields on the envelope to form black patches.

5. As the redundant information such as the sender’s address and the stamp impressions are usually present only in the corner of an envelope, they can be removed by scanning through the image.

6. The resultant image now has a white back ground with back patches only in the destination address.

7. After leaving a tolerance, a rectangle is suitably positioned over the destination address patch and its coordinates are returned.

* 1. **Flowchart of RC algorithm**



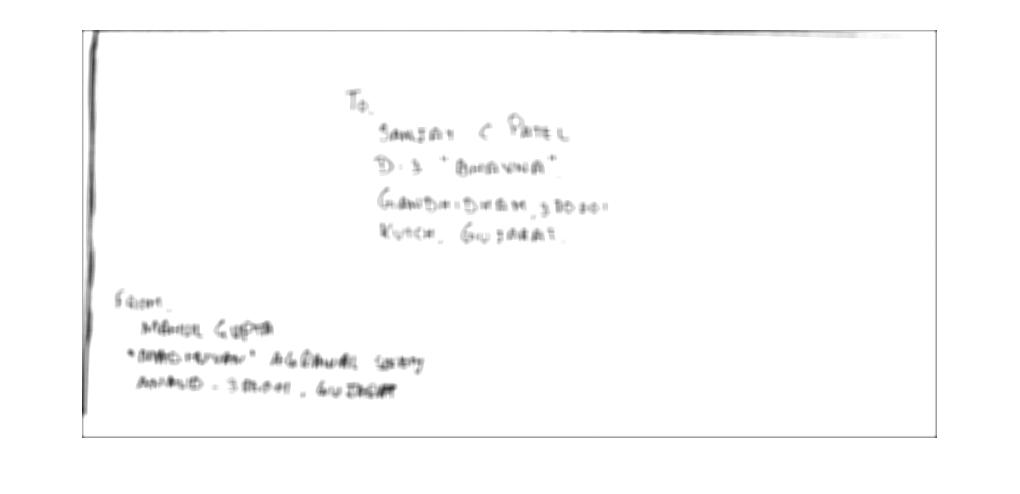
*Figure 12: Redundancy Correction Algorithm*

* 1. **Implementation of Redundancy Correction Algorithm**

The original image in Figure 11 is Low Pass Filtered using a 20 x 20 mask spatial mask. The spatial mask is nothing but a matrix with 20 rows and 20 columns with all of its elements set to 1. A 20 x 20 mask is chosen as it provides sufficient blurring effect which can be converted into patches by re-Binarization. The Low Pass Filtering is accomplished by rolling the 20 x 20 mask through each pixel of the original image and determining its response. The value of the pixel is subsequently transformed into the response value.

*R = 0.00285 \* (x1 + x2 + x3 + x4……………………………. + x17 + x18 + x19 + x20)*

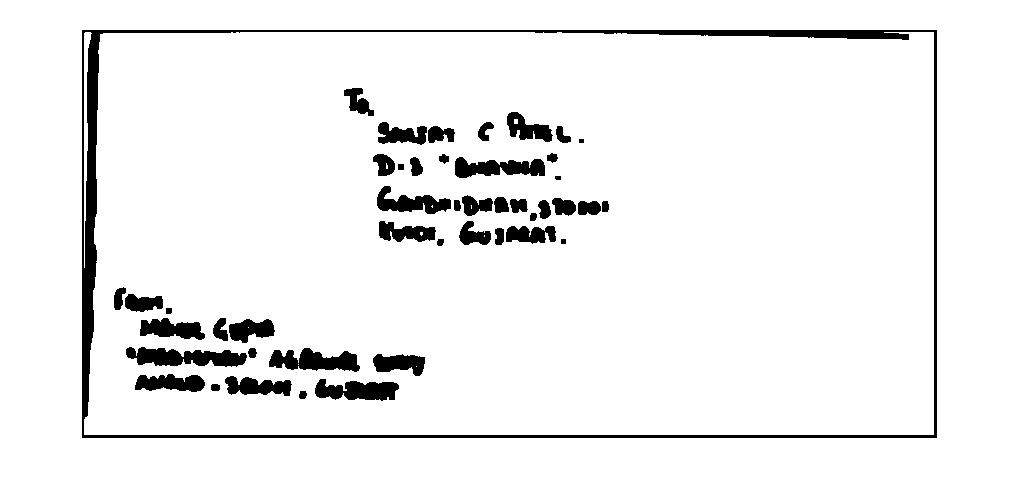
The equation for computing the response by point processing is given above. In this equation x1, x2… x20 represent the value of the pixels of the original image over which the 20 x 20 spatial mask is placed.



*Figure 13: Low Pass Filtered Image*

It can be clearly observed from the image that the Low Pass Filtering has blurred it sufficiently for patch formation. Another important point to be noted here is that the size of Figure 13 is around eight times the size of Figure 10 as 8 bits are required representing each pixel of Figure 13.

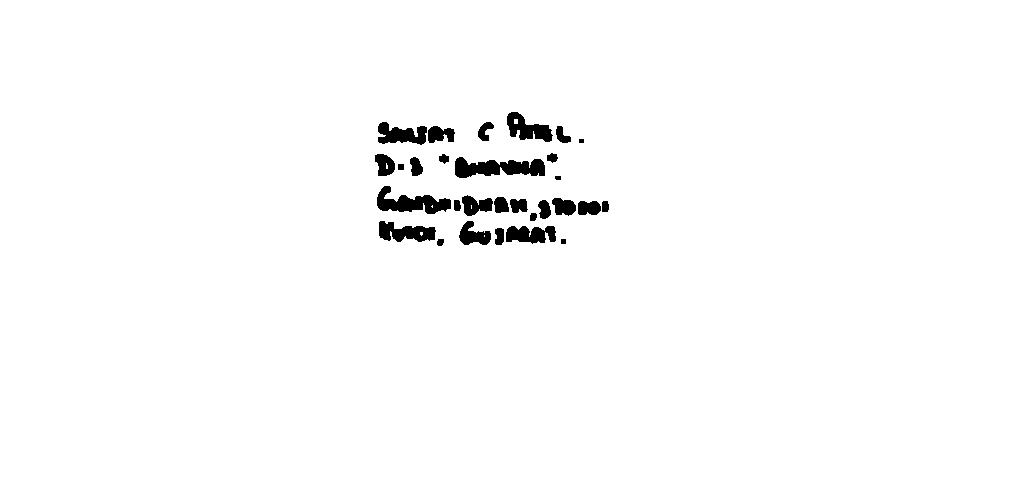
The next step is to re-binarize the Low Pass Filtered image (Figure 13) to obtain an image with patches. The degree of patch formation can be adjusted by varying the threshold level during binarization.



*Figure 14: Re – Binarized Image*

In our application, it is desirable to form continuous patches so that all the words are joined together. Consequently, a high threshold value of the order of (0.8 to 1.0) is set during binarization .The above Binary image (Figure 13) is obtained by using a threshold of 1.0

The Redundancy Corrected image (Figure14) is obtained from Figure 13 by removing the unwanted information in the corners and boundaries of the image. As it is conventional that the Destination address is written somewhere in the centre and certainly not in the corners, such logic has been employed.



*Figure 15: Redundancy Corrected image*

After redundancy correction, the image is left only with the patches of the destination address. By horizontal scanning from the first pixel row, the location of the Destination address can be accurately identified. A tolerance level is adopted and a rectangle is drawn over the destination address. The coordinates of this rectangle is returned to the Line Segmentation Module.

1. **Segmentation Techniques**

**9.1 Pixel Approach**

The line separation procedure consists of scanning the image (Figure 15) row by row. The row in the preceding line represents the pixel row and not the lines of the address. The simplest way of separating the lines is to set a threshold value for the number of white pixel rows between two address lines. Two lines are separated if the number of white pixel rows between them is greater than the threshold value. Such a logic would be futile when letters such as ‘y’,’g’ etc occur in the first line and letters like ‘f’,’d’, etc occur in the second line without having white pixel rows in between. Such a bottle neck can be averted by designing the algorithm in such a way that it is tolerant to a certain minimum number of black pixels in a white row.

**9.1.1 The pixel approach Algorithm:**

1. Scan the extracted image (Figure 16) pixel-row by pixel-row.

2. Set a threshold value for the minimum number of white pixel rows to be present between two address lines.

3. A predefined amount of tolerance is provided while deciding whether a pixel-row is a white pixel-row for avoiding ‘y’ – ‘f’ problem.

4. Count the number of consecutive white pixel rows.

5. If the number of white pixel rows is greater than the threshold, then separate the two address lines.

6. Repeat steps 4 and 5 until all the address lines have been separated.

**9.2 Histogram approach**

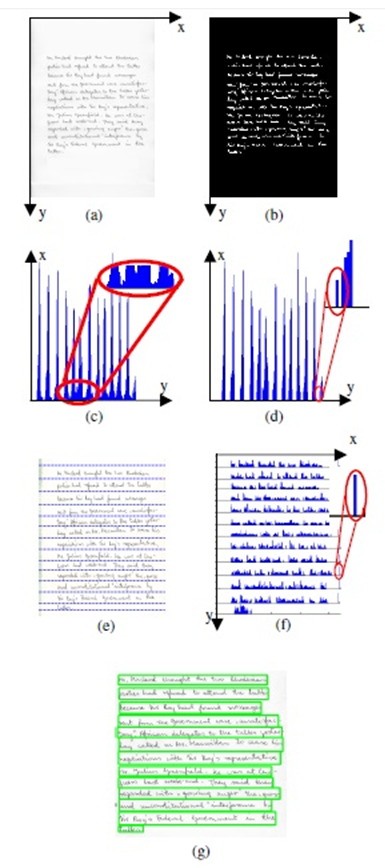
Histogram approach is a method to automatically identify and segment the text line regions of a handwritten document. The feature extraction or binarization step is applied to the input image, resulting in Figure 16b. Then, a Y histogram projection (Figure 16c) is obtained to detect the possible lines. Due to some noises, a text line separation (Figure 16d) is necessary. Once the false lines are found, they must be excluded. After that, the line region recovery step (Figure 16e) is performed in order to recover some losses introduced by the preceding step. An X histogram projection that is applied to each line detected (Figure 16f) takes out possible false words, mainly at the lateral edges of the page. Finally, we obtain the text lines region (Figure 16g).

**9.2.1 Y Histogram Projection**

Once the feature extraction of the images is performed, the Y histogram projection of the whole image is obtained. The idea is to use a simple and fast method to correctly distinguish possible line segments in the handwritten text. In Figure 16c it is clear that each text line corresponds to a peak in the histogram. The histogram represents the added pixels for each y value. So the empty spaces between the peaks represent possible regions between different text lines.

**9.2.2 Text Line Separation [30]**

Once all the potential lines are detected, a procedure to apply a threshold is performed to obtain a possible line separation in the text. This threshold is dynamically calculated and it is proportional to the average length of the lines in the text (Y histogram values). This process applied to the histogram aims to remove the regions in the histogram that are not referred to the lines in the text, or the elimination of noises that confuses with the text lines. The choice of the parameter to be used as threshold is intrinsic related to the information in the text, so that the algorithm utilized the minimum possible of heuristic techniques to determine the line separation points. Actually, this stage tries to identify the location of each text line. The separation of the possible text lines regions using the histogram shows a difficult due to the upper and lower regions of some letters as shown in Figure 17.



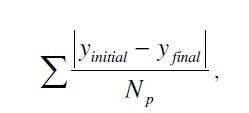
*Figure 16: Histogram Method [29]*

****

*Figure 17: Region that provoke false lines [29]*

**9.2.3 False Line Exclusion**

This procedure tries to exclude possible noises close to the text lines regions. Once the possible text line regions are separated by removing an offset from the histogram, we determine the average height of these regions to exclude false lines that might be detected. In Figure 16d we can observe this effect, a small peak in the histogram shown in red ellipse. If this region poses enough height it can be confused with a text line segment by the algorithm. The height of a line is obtained by taking the limit values of the corresponding region in the Y histogram and calculating the difference between them. The equation bellow provides the average height of the lines found in a page:



Where, yinitial is the y position where the text region begins, yfinal is the y position where the text region ends and Np is the number of regions found in the page. The lines with height below a pre-determined threshold are removed. The value of this threshold is proportional to the average height of the text lines in the whole image.

**9.2.4 Line Region Recovery**

This procedure determines the average point between the regions found. The idea is to find the maximum area that each line might be inscribed, by determining the superior and inferior coordinates in the y axis. Figure 16e shows the limits of these regions after the exclusion threshold is applied. The dashed lines are the limits between two adjacent line regions. In this way, the excluded regions are recovered. Note that the limit lines establish the maximum and minimum y coordinates for each text line.

**9.3 Smearing Method**

In smearing methods, the consecutive black pixels along the horizontal direction are smeared consequently; the white space between black pixels is filled with black pixels. It is valid only if their distance is within a predefined threshold. This way, enlarged areas of black pixels around text are formed. It is so-called boundary growing areas. These areas of the smeared image enclose separated text lines. Hence, obtained areas are mandatory for text line segmentation.

**9.4 Hough Transform Method [14]**

The Hough transform [Ballard, 1981] is a widespread technique for finding straight lines in the images. Consequently, image is transformed in the Hough domain. Potential alignments are hypothesized in Hough domain and validated in the image domain. Apart from that, projection profile and Hough transform are identical methods in principle. Projection profile shows more variation when the profile is in the direction to the document skew. The direction for the maximum variation is then determined by a cost function. Similarly, the “voting” function in Hough domain determine slope of the straight line [Amin and Fischer, 2000].

**9.5 Stochastic Method [2]**

Stochastic method is based on probabilistic algorithm, which accomplished nonlinear paths between overlapping text lines. These lines are extracted through hidden Markov modelling (HMM). This way, the image is divided into little cells. Each one them correspond to the state of the HMM. The best segmentation paths are searched from left to right. In the case of touching components, the path of highest probability will cross the touching component at points with as less black pixels as possible. However, the method may fail in the case that contact point contains a lot of black pixels.

**9.6 Water fall Method** [2]

**9.6.1 Basic Water Flow Algorithm**

Original water flow algorithm proposed, assumes hypothetical water flows under a few angles of the document image frame from left to right and vice versa. In this hypothetically assumed situation, water is flowing across the image frame. For the water flows from left to right, the situation is shown in Figure 18. Areas that are not wetted form unwetted ones. The stripes of unwetted areas are labelled for the extraction of text lines. Further, this hypothetical water flow is expected to fill up the gaps between consecutive text lines. Hence, unwetted areas left on the image frame lies under the text lines. Once the labelling is completed, the image is divided into two different types of stripes. First one contains text lines. The other one contains line spacing.



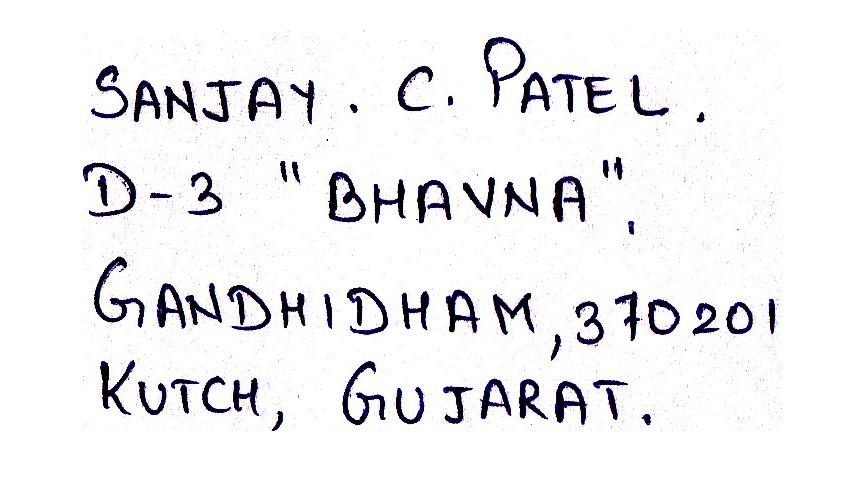
*Figure 18: Unwetted area definition [2]*



*Figure 19: United unwetted areas [2]*

1. **Line Segmentation Module**

As stated earlier a copy of the original image (Figure 11) is stored separately as the ABL module distorts the image given to it. The Coordinates returned by the ABL module is used to extract the destination address from the saved copy of Figure 11. The extracted destination address is shown below.



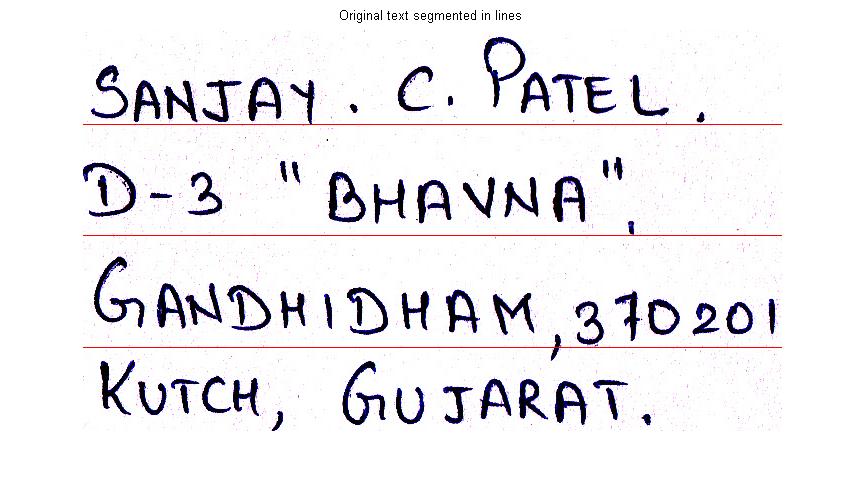
*Figure 20: Extracted Address*

* 1. **Line Segmentation Logic and Constraints**

From [30], we can draw the conclusion that pixel approach is the most suitable method as per our requirements. Here we have assumed that the size of each font is nearly same, also the space between two lines is equal for any set of lines, The following shows line segmentation logic.

The line separation procedure consists of scanning the image (Figure 20) row by row. The row in the preceding line represents the pixel row and not the lines of the address. The simplest way of separating the lines is to set a threshold value for the number of white pixel rows between two address lines. Two lines are separated if the number of white pixel rows between them is greater than the threshold value.

Such a logic would be futile when letters such as ‘y’,’g’ etc occur in the first line and letters like ‘f’,’d’, etc occur in the second line without having white pixel rows in-between. Such a bottle neck can be averted by designing the algorithm in such a way that it is tolerant to a certain minimum number of black pixels in a white row. In other words, a row with very few black pixels can be considered as a white pixel row. Such an algorithm will be fault tolerant.

****

*Figure 21: Line separation.*

**10.2 Horizontal Scanning (HS) Algorithm**

The Horizontal scanning algorithm that is used for line separation is described below.

1. Scan the extracted image (Figure 20) pixel-row by pixel-row

2. Set a threshold value for the minimum number of white pixel rows to be present between two address lines.

3. A predefined amount of tolerance is provided while deciding whether a pixel-row is a white pixel-row for avoiding ‘y’ – ‘f’ problem.

4. Count the number of consecutive white pixel rows.

5. If the number of white pixel rows is greater than the threshold, then separate the two address lines.

6. Repeat steps 4 and 5 until all the address lines have been separated.

**10.3 Flow Chart for HS Algorithm**

The Flowchart given for the line separation using horizontal scanning algorithm will work perfectly only when the address is written in a horizontal manner. The tolerance value EL has to be set for avoiding high precision errors. In other words, while scanning a horizontal pixel row, a white row need not mean that all the pixels in it are white. A maximum of EL number of pixels can be black in colour. By using such a tolerance, even if few characters of two address rows cross a same pixel row, line separation algorithm would work efficiently.

1. **Future Enhancements**

The Automatic Mail Processor, whose design and working was explained in the preceding pages, is practically implementable. Cost efficiency is ensured as only barcode readers are required at sorting stations. The speed of operation would be exponentially high when compared to manual sorting. As everything is automated, the chances for errors in interpretation are very less.

Though every module of this project completely serves its purpose, enhancements can be made in the recognition process we have used. Instead of using only the outer contour of each character, the inner contour can also be taken into consideration. If the Back Propagation Neural network is replaced by an Adaptive network like ART, then the training process for new samples would not be cumbersome.

So far Segmentation module has been successfully implemented. The next thing to be performed are character recognition, address parser, and bar code generation. We also plan to propose a prototype for the same.

We are implemented this project for the English language. But for real use, this must be extended to other Indian languages too. The process of extending this project to other languages is not difficult as the recognition process undertaken here is independent of the language used. The Neural Network used in the final stage of recognition should be trained with character patterns of other Indian languages too.

D:\COLLEGE\PROJECTS\Final year projects\Mid sem report\Report\images\11.tif

*Figure 22: Flow Chart of HS Algorithm*

**12 References**

|  |  |
| --- | --- |
| [1] | Nafiz Arica and Fatos T. Yarman-Vural ,"An Overview of Character Recognition Focused on Off-Line Handwriting" in IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS,May 2001 |
| [2] | Darko Brodić and Zoran Milivojević ,"A New Approach to Water Flow Algorithm for Text Line Segmentation" in Journal of Universal Computer Science, vol. 17, no. 1,2011 |
| [3] | J. Serra, “Morphological filtering: An overview,” Signal Process., vol.38, no. 1, pp. 3–11, 1994. |
| [4] | R. Legault and C. Y. Suen, “Optimal local weighted averaging methods in contour smoothing,” IEEE Trans. Pattern Anal. Machine Intell., vol.18, pp. 690–706, July 1997. |
| [5] | S. Mo and V. J. Mathews, “Adaptive, quadratic preprocessing of document images for binarization,” IEEE Trans. Image Processing, vol. 7, pp. 992–999, July 1998. |
| [6] | W. L. Lee and K.-C. Fan, “Document image preprocessing based on optimal Boolean filters,” Signal Process., vol. 80, no. 1, pp. 45–55, 2000. |
| [7] | A. Polesel, G. Ramponi, and V. Matthews, “Adaptive unsharp masking for contrast enhancement,” in Proc. Int. Conf. Image Process., vol. 1,1997, pp. 267–271. |
| [8] | J. Serra, “Morphological filtering: An overview,” Signal Process., vol.38, no. 1, pp. 3–11, 1994. |
| [9] | J. M. Reinhardt and W. E. Higgins, “Comparison between the morphological skeleton and morphological shape decomposition,” IEEE Trans.Pattern Anal. Machine Intell., vol. 18, pp. 951–957, Sept. 1996. |
| [10] | J. Yang and X. B. Li, “Boundary detection using mathematical morphology,” Pattern Recognit. Lett., vol. 16, no. 12, pp. 1287–1296, 1995. |
| [11] | J. Kanai and A. D. Bagdanov, “Projection profile based skew estimation algorithm for JPIG compressed images,” Int. J. Document Anal.Recognit., vol. 1, no. 1, pp. 43–51, 1998. |
| [12] | A. Hashizume, P. S. Yeh, and A. Rosenfeld, “A method of detecting the orientation of aligned components,” Pattern Recognit. Lett., vol. 4, pp.125–132, 1986. |
| [13] | M. Chen and X. Ding, “A robust skew detection algorithm for gray-scale document image,” in Proc. 5th Int. Conf. Document Anal. Recognit.,Bangalore, India, 1999, pp. 617–620. |
| [14] | G. Louloudis1, B. Gatos2, I. Pratikakis2, C. Halatsis1,"Line And Word Segmentation of Handwritten Documents". |
| [15] | E. Oztop et al., “Repulsive attractive network for baseline extraction on document images,” Signal Process., vol. 74, no. 1, 1999. |
| [16] | Y. Tang et al., “A new approach to document analysis based on modified fractal signature,” in Proc. 3rd Int. Conf. Document Anal. Recognit.,Montreal, QC, Canada, 1995, pp. 567–570. |
| [17] | F. Cesarini, M. Gori, S. Mariani, and G. Soda, “Structured document segmentation and representation by the modified X–Y tree,” in Proc. 5th Int. Conf. Document Anal. Recognit., Bangalore, India, 1999, pp.563–566. |
| [18] | T. Pavlidis and J. Zhou, “Page segmentation by white streams,” in Proc. 1st Int. Conf. Document Anal. Recognit., Saint-Malo, France, 1991, pp.945–953. |
| [19] | O. Akindele and A. Belaid, “Page segmentation by segment tracing,” in Proc. 2nd Int. Conf. Document Anal. Recognit., Japan, 1993, pp.314–344 |
| [20] | S. Tsujimoto and H. Asada, “Major components of a complete text reading system,” Proc. IEEE, vol. 80, p. 1133, July 1992. |
| [21] | L. D. Harmon, “Automatic recognition of print and scripts,” Proc. IEEE, vol. 60, pp. 1165–1177, 1972. |
| [22] | R. G. Casey and G. Nagy, “Recursive segmentation and classification of composite patterns,” in Proc. 6th Int. Conf. Pattern Recognit.,München,Germany, 1982, pp. 1023–1031. |
| [23] | K. Fukushima and T. Imagawa, “Recognition and segmentation of connected characters with selective attention,” Neural Networks, vol. 6, no.1, pp. 33–41, 1993. |
| [24] | E. Gilloux, “Hidden Markov models in handwriting recognition,” in Fundamentals in Handwriting Recognition. ser. NATO ASI F, S. Impedovo,Ed. New York: Springer-Verlag, 1994, vol. 124. |
| [25] | C. Chen and J. DeCurtins, “Word recognition in a segmentation-freeapproach to OCR,” in Proc. 2nd Int. Conf. Document Anal. Recognit.,1993, pp. 573–579. |
| [26] | S.W. Lee, D. J. Lee, and H. S. Park, “A new methodology for gray-scale character segmentation and recognition,” IEEE Trans. Pattern Anal. Machine Intell., vol. 18, pp. 1045–1050, Oct. 1996. |
| [27] | Y. Hamamoto et al., “Recognition of handprinted Chinese characters using Gabor features,” in Proc. 3rd Int. Conf. Document Anal. Recognit.,Montreal, QC, Canada, 1995, pp. 819–823. |
| [28] | Naf Iz Arica,"An off lline character recognition system for free style handwriting" , in the middle east technical university. |
| [29] | Rodolfo P. dos Santos, Gabriela S. Clemente, Tsang Ing Ren and George D.C. Calvalcanti,"Text Line Segmentation Based on Morphology and Histogram Projection",in 10th International Conference on Document Analysis and Recognition,2009. |
| [30] | Pulagam Soujanya, Vijaya Kumar Koppula, Kishore Gaddam & P. Sruthi,"Comparative Study of Text Line Segmentation Algorithms on Low Quality Documents",in CMR College of Engineering and Technology Cognizent Technologies, Hyderabad, India |