Literature review

Research project: Handwriting English Recognition

OUGUNIR Safaa

Abstract

This work aims to provide a review of research on English handwriting recognition technology. Handwriting recognition is a difficult task since people have individual handwriting styles, the text cannot be supposed to be written in straight lines, neighboring characters within a word are usually connected and we may need to segment a word into separate characters for accurate character recognition. In the recent years, many methods and algorithms have been proposed; nevertheless, the problem remains one of the most challenging problems in pattern recognition and even now there is no single approach that solves it both efficiently and completely. Thus, this review will be organized as follows: the first section will present the prior OCR phases: preprocessing, line segmentation. The second section will review the difference between offline and online handwriting recognition in order to compare different methods and their difficulties. The third section will focus on character segmentation techniques. The last section will discus methods that have been proposed for character recognition.

Keywords: HWR (Handwriting recognition), handwriting English text recognition, line segmentation, word segmentation, beautification, neural networks

Table of Contents

List of Figures

Introduction

Handwriting Recognition is even hard for the human eye; reading in absence of context fails 4% of the times [1]. Thus, handwritten text recognition is a very challenging problem. Over the last few years, Significant progress has been made in handwriting recognition technology. Thus far, HWR systems still incomplete. In fact, the large majority of these existing systems have been limited to small and medium vocabulary applications, since most of them are often relying on restricted vocabulary called lexicon during the recognition process. They are only capable of recognizing words that are present in a list of words called lexicon. As such, the success of recognizing will always depend on the lexicon size and as the number of words grows the more difficult the recognition becomes. So even in online handwriting recognition the size of vocabulary poses serious challenge and researchers avoid using large number of words [2]. This review focuses mainly on an open vocabulary system, i.e. system that are capable to dealing with any sequence of handwringing text without relying on a lexicon. This system is mainly based on five phases, namely preprocessing, line segmentation, beautification, character segmentation and character classification. Preprocessing aims to reduce noise and distortion, baseline correction so that to produce an input data easy for OCR system to work accurately. Word segmentation of lines then Character segmentation words into a sequence of a basic recognition units (which are in our case characters). Character classification consists of pattern training in order to predict the unknown characters.

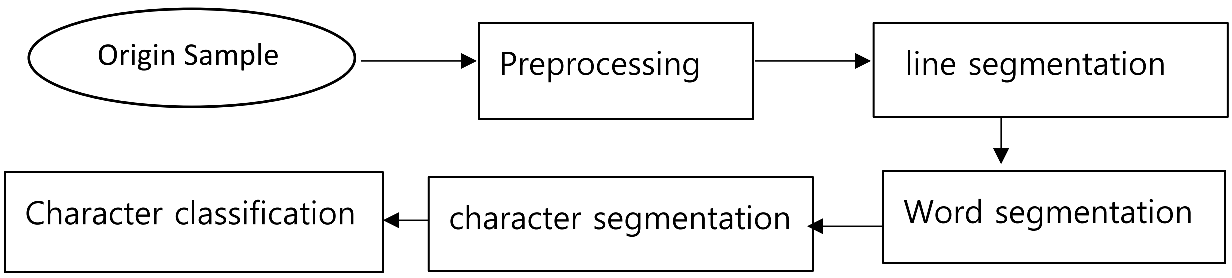


Figure 1. The basic architecture of the proposed offline handwriting recognition system

1. Phases of OCR

a) Preprocessing

It is necessary to make some document analysis operations prior in order to organize the information so that the subsequent recognition process become simpler. The common various operations performed on the image during pre-processing stage are discussed differently in many articles ([1] [2] [3] [4] [5] [6] [7] [8]) but still have the same purpose: **Detection**, Reading the scanned image and cropping the image to find the area of interest; **thresholding**, the fact of converting a gray-scale image into a binary black-white image; **noise removal**, Removing noise from the image; **beatification**, Skew Detection and Correction.

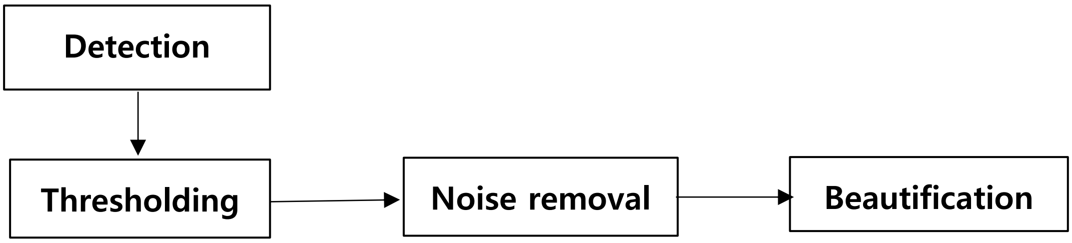


Figure 2. Preprocessing Tasks

- **Detection**

Localizing text in images vary a lot depending on the nature of image. Many approaches have been proposed for text localization in natural images (also known as text-in-the-wild problem) [9] [10]. In [9], The proposed approach for text region detection is based on edge detector operator such as canny edge detector which is used to detect wide range of edges in images. The Canny edge detector uses multistage algorithm to obtain sharp edges in the image and it's used after grayscaling and removing noise from the input natural scene image. Another approach was proposed in [10], Stroke width transform based on clustering techniques (unsupervised learning) is a local image Operator which calculates per pixel the width of the most likely stroke containing the pixel [10]. The output of the SWT is an image of size equal to the size of the input image where each element contains the width of the stroke associated with the pixel. They defined stroke to be a contiguous part of an image that forms a band of a nearly constant width. The text is then distinguished under the hypothesis that the textual elements are writing with homogenous stroke width and may therefore be detected using clustering techniques. In this field, [12] have proposed a new method based on maximum color difference and boundary growing methods for detecting multi-oriented handwritten text in video which is also useful for images. The proposed method was tested on publicly available data (Hua's data) and it works well. It was able to detect small and low contrast handwritten text in video.

- **Thresholding**

The thresholding (also called binarization) extracts the foreground of text from the background [11] [7] [5]. It aims to change the grayscale image into binary image by removing the background. Here, we want to generate an image where the background pixel values are equal to zero and the foreground pixel values are equal to 255. So, to classify each pixel as foreground or background, we make two general limitations of handwritten text [7]:

1. Handwritten text is written with a dark color on a bright background
2. Areas containing mainly foreground pixels have a larger standard deviation of intensity values than areas containing mainly background pixels.

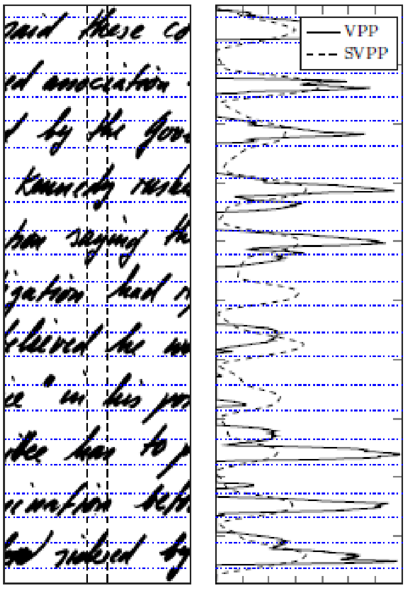
Thus, thanks to these assumptions the histogram of the input image will consist of two peaks: a high peak matching the bright background and a smaller peak matching the foreground. So, determining an" optimal" threshold gray-scale value between the two peaks is one of important tasks before the binarization value [5].

- **noise removal**

Noise is introduced during the acquisition of pictures via optic scanners or devices of writing and cause a distortion in the input image. Before recognition of the text, it is essential to remove these variants. Although, there are many techniques used to eliminate. Filtering approaches is the most used. It aims to remove noise and diminish spurious points, usually introduced by uneven writing surface and/or poor sampling rate of the data acquisition stage [8]. A range of spatial and frequency domain filters can be designed for the noise removal. The basic idea is to convolute a predefined mask with the image to assign a value to a pixel as a function of the gray values of its neighboring pixels. Filters can be conceived for smoothing, sharpening, thresholding, removing slightly textured or colored background, and contrast adjustment purposes [8]. Another approach which is also widely used is Morphological Operations. Its basic idea is to filter the document image substituting the convolution operation by the logical operations. Many morphological operations can be designed to connect the broken strokes, decompose the joined strokes, smooth the contours, reduce the wild points, thin the characters, and extract the boundaries. Then, morphological operations can be used successfully to remove the noise from the document images, due to poor quality of ink and document, as well as inconsistent hand movement [8].

- **Beautification (Skew and slant correction)**

Beautification aims to reduce the variance in characters position in order to line them in one baseline and make them more legible. Baseline correction is the first task in this part. It's a process that decreases the variance in the relative position of characters with respect to their neighbors along a text line. This part consists on detecting the baseline of the text line and transforming the image so that founded baseline becomes horizontal. The Baseline is an imaginary horizontal line which links up the characters of a word. The detection of the baseline is one of the important tasks in the preprocessing. it's useful in slanting correction as well as line segmentation. A common method used to extract the baseline is vertical Projection Profile and it works well. The VPP is done by projecting the image onto the vertical axis, which is to sum the pixels values in each row. The result shows different peaks where the text lines are. Here are some results from [11]:



(a) Investigated vertical segment. (b) The VPP and SVPP of the segment.

Figure 4. VPP method

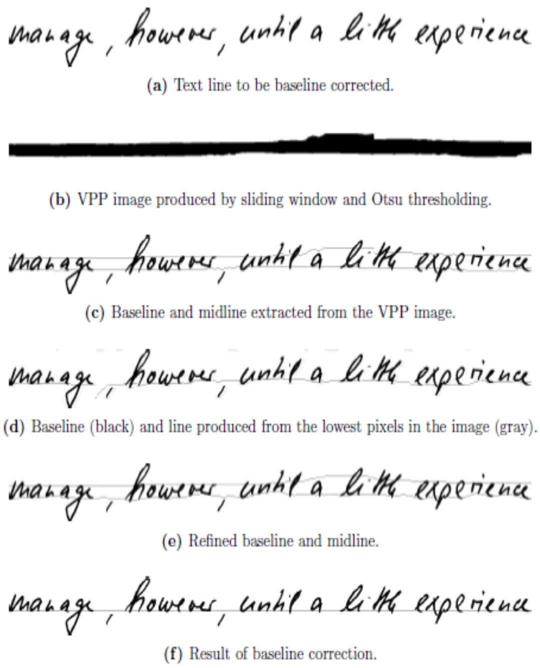
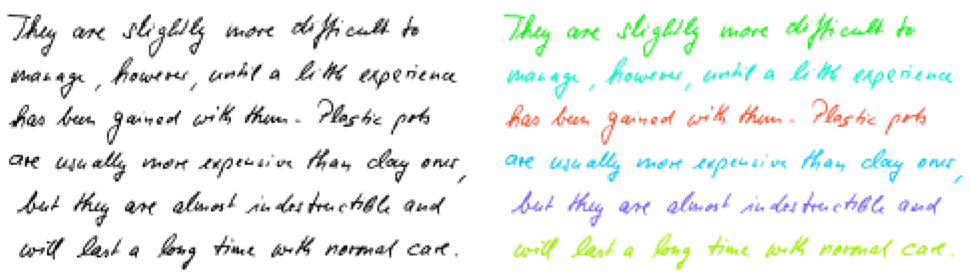


Figure 3. Line Segmentation procedure

The baseline correction method is based on this method vertical translation. For each row of the VPP image there might be more than one area of following foreground pixels. Therefore, the longest chain of foreground pixels in each column will correspond to the corpus of the text line. By finding the highest and lowest pixel of the corpus, we now have an estimate of the midline and baseline, as shown in Figure c. The baseline is then refined by comparing it with a line produced by finding the lowest foreground pixel in a small neighborhood, as shown in Figure d. The average distance between the midline and baseline provides a measure of the character height, which is used to refine the baseline, which is illustrated in Figure e. The distances for each point of the estimated baseline from the average y-position of the baseline are then used to translate each pixel column vertically in order to correct the baseline, as shown in Figure f.
Using the bounding boxes of the text lines, the position of the text lines can be adjusted so that the left side or the center of the bounding boxes are aligned. The new left padding of the bounding boxes is based on the smallest distance to the left margin for left alignment and half of the difference between the image width and the bounding box width for center alignment.

b) Line segmentation

Line segmentation is the separation of individual lines of text, in order to facilitate other levels of segmentation. Line segmentation is done using Horizontal Projection Profile. In the histogram a text documents a peaks and valleys represents the presence of text and space between lines respectively.
Here are the results of the text line segmentation performed by the Vertical Projection Profile Method In [11]. It demonstrates an example of segmented text lines written with different colors.



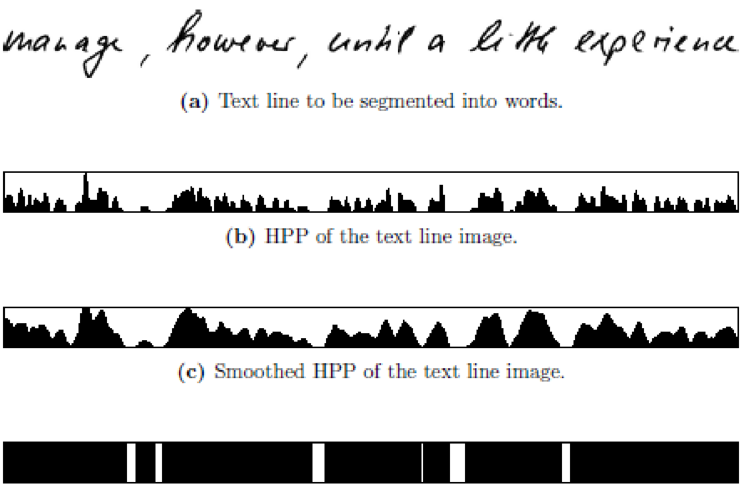
(a) Binary image to be segmented. (b) Segmented text lines in different colors.

Figure 5. Line segmentation results using HPP

2. Word segmentation

After the lines segmentation, we are going to try to segment every line of the text into words. The words segmentation depends on the gaps between words. However, some characters are not necessarily connected to a following character of a word, what creates other gaps between sub-words. The longer areas separate words while the short areas separate sub-words. As a result, most researchers assume in their systems that the gaps between words are larger than gaps between sub-words. The most used approach for this phase is the Horizontal Projection Profile (HPP) of the text line image. It starts by projecting the image on the horizontal axis, which is taking the sum over the pixel values in each column; A Smoothed HPP (SHPP), is obtained by smoothing the HPP using an averaging filter of width, chosen so that the spaces within the words are closed but leaves the gaps between the words open. [11]. We get an image that reveals the limits of each word.

Here are the results achieved in [11] for the line segmentation using HPP:



(a) Text line to be segmented into words.

(b) HPP of the text line image.

(c) Smoothed HPP of the text line image.

(d) Thresholded smoothed HPP using Otsu's method.

Figure 6. Steps in the word segmentation process.

3. Character segmentation:

The Character segmentation aims to split the images of handwritten words gotten from the last phase into pieces corresponding to single characters. Many segmentation methods have been developed by the researchers in the last few years. The main problem in segmenting handprinted words is to split touching characters. When a word image is handwritten, connection between adjacent characters may happen due to style of writing especially for lower case characters . A stroke belonging to a character may touch a stroke of the adjacent one in some cases. In this section, we discuss many different types of algorithms for segmenting handwriting characters. The majority of the methods are based on oversegmentation – finding some set of possible splitting points in the graphical representation of the word and then trying to eliminate the improper ones.

A fuzzy approach to segment touching characters had been proposed in [16]. this proposed approach defines a function based on features that describe a touching position. Then, such a function is evaluated for each column of the matrix in order to choose the cut position. The cut position (or cutting column) is the number of the column of the matrix chosen for separating the pattern into two sub–patterns. The strategy had been tested for Latin handwriting text and had shown good results.

Another New Character Segmentation Approach in [17] is based on the analysis of the ligatures in an accurate manner by calculating the height and width of the word image. In fact, the word image is scanned vertically, from top to bottom, column wise and the number of foreground pixels in the inverted word image are counted in each column. The locations of all these columns are saved for which the sum of foreground black pixels is either 0 or 1. All these identified columns are termed as PSC (Potential Segmentation Columns) as shown in Fig. 7 (d).

Figure 7. Word Image Segmentation (a) Pre-processed Word Images; (b) Inverted Binary Images; (c) RGB Images; (d) Over-segmentation in

Other researches had proposed a loop between the segmentation and classification phases in order to increase the system accuracy [19]. The idea is to split words into segments that should be characters, pass each segment to a classifier and, if the classification results are not satisfactory, call segmentation once more with the feedback information about rejecting the previous result. Eventually, only the most plausible segmentation points will be selected with the support of a classifier. In the latter case the tentative evaluation of the candidate points at the segmentation step is useful to find the order in which the points should be presented to the next step. In all those cases decision on which candidate split points will be finally used is taken together by a segmentation and classification module. The last case is employed to great extent in recent publications. Main steps of the methods defined as the last case above take often the form of: 1. find many possible break points, 2. tentatively evaluate those points, 3. remove those that are very unlikely to be the correct break points and, if necessary, add missing ones, 4. present the candidate points to the classification step in some order taking into account their initial evaluation.

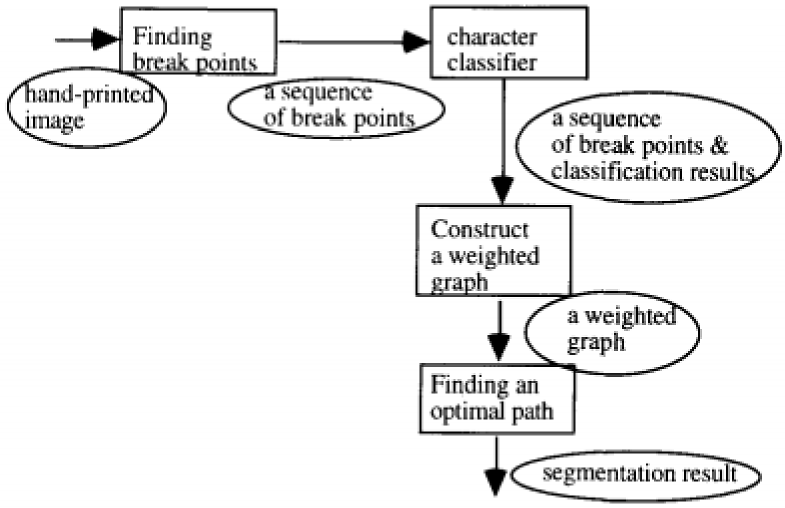


Figure 8. Control of segmentation and recognition of handwriting words

In the same filed, the [20] paper presented a character segmentation method for cursive handwritten text Using Transformation-Based Learning. This method is used in order to extract automatically rules that notice the segment boundaries. The experimental results for this research had achieved 82 % of accuracy in detecting segment. The segmentation process has two phases. Initially, a first estimation of the segment boundaries is done using a simple algorithm. The concept of transformation-based learning (is a machine learning) is, then, used in order to decide, on the basis of disambiguation rules extracted in a training phase, which segment boundaries, proposed by the presegmentation module, are actual segment boundaries. The method of TBL is a machine learning theory that needs already annotated data for training in order to extract the appropriate knowledge and convert it to simple useful rules. The rules are of the following format: IF triggering environment THEN transformation where the transformation changes the state of a tag (e.g., a segment boundary) if the condition defined in the triggering environment is valid. Initially, the training data are annotated based on an initial-state annotator. The extraction of rules is done via an iterative process. During each iteration, all the possible transformations are tested and the one that achieves the best results (by comparing the derived data with the already manually annotated data) is selected [20].

4. Character recognition (classification)

The classification phase is the most important part of the English handwriting recognition system. It seeks to classify an individual handwritten character into 26 classes or more. The performance of a classifier depends on the quality of the input segments. There are several existing techniques for handwriting classification, it can be studied in two classes. They are specified in [8] as :

(a) Classical Techniques: Template matching, Statistical techniques, Structural techniques.
(b) Soft Computing Techniques: Neural networks (NNs), Fuzzy- logic technique, Evolutionary computing techniques.

1) Template matching: It is based on matching the existing stored models against the segmented pattern to be recognized, in our case it will be an unknown character. In General, Matching operation determines the degree of similarity between two vectors (group of pixels, curvature, shapes etc.) in the feature space.

2) Statistical Techniques: Statistical decision theory is based on statistical decision functions and a set of optimality criteria, which maximizes the likelihood of the unknown pattern given the model of a certain class. These techniques are based on three major assumptions [8]:

a) Feature set distribution is Gaussian or in the worst-case uniform,

b) There are enough statistics available for each class, {1,...., n}, which represents

c) Given a group of images {I}, one is able to extract a set of features {fi} ∈ F, i ∈ each distinct class of patterns. The measurements taken from n-features of every unit can be thought to represent an n-dimensional vector space and the vector, whose coordinates correspond to the taken measurements, shows the original unit.

3) Structural Techniques: The recursive description of a complex pattern in terms of simpler patterns based on the shape of the object was the initial idea behind the creation of structural pattern recognition. These patterns are used to describe and classify the characters in the CR systems [8].

4) Neural Networks (NN):

For several years, neural networks, and particularly multi-layered perceptron, had shown great result in the field of recognition statistics of forms. These systems are composed of a group structured by units of treatment, called 'neurons'. These units are working in parallel and are strongly interconnected.

Because of its parallel nature, it can achieve calculations at a higher rate compared to the traditional techniques. It's dynamic and can easily adapt to changes in the data and learn the input indicators characteristics. In fact, a neural network contains many nodes. The output from one node is input to another one in the network and the final decision depends on the complex collaboration of all nodes. Several approaches exist for training of neural networks [26]. These include the boltzman, error correction, hebbian and competitive learning. They cover binary and continuous valued input, in addition with supervised and unsupervised learning. On the other hand, neural network architectures can be classified into two major groups, namely, feed-forward and feedback (recurrent) networks. The most familiar neural networks used in the CR systems are the multilayer perceptron of the feed forward networks and the kohonen's Self Organizing Map (SOM) of the feedback networks [8]. Finally, networks of large are also very demanding in power of calculation, what restrains strongly their possibilities of application.

Here are some proposed recognition systems based on neural networks in many studies [23] [22] [3]:

The [22] pepper proposed a neural network architecture based on geometric characteristics of letters. After the preprocessing and segmentations phases, the next step is to find the center of mass of the character image. With the center of mass as a reference point, the vectors are drawn, creating a set of points composing the contour of the character so that its pattern description is made. The neural network training patterns are based on the geometric analysis of characters. The description patterns of each isolated character are inputs for an artificial neural network (fig 9).

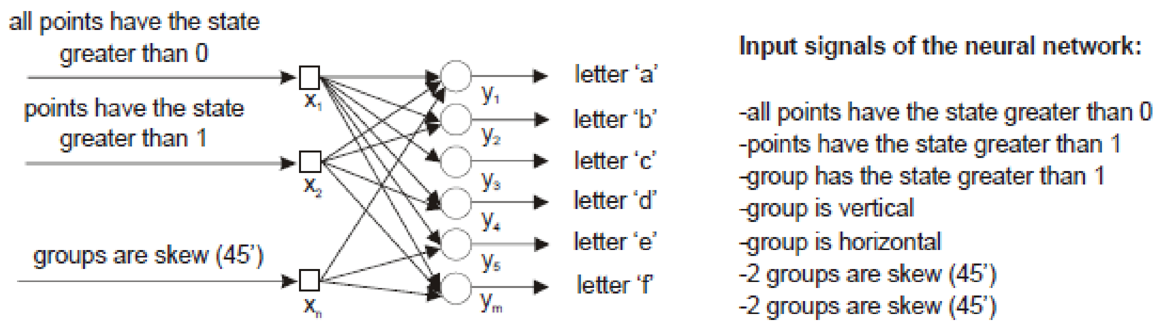


Figure 9. Input signals for the neural network proposed system

In the same field, a Convolutional Neural Networks system was proposed in [3]. The proposed Architecture of CNNs is the common model the LeNet-5 model (fig 10). Each unit in it is connected to a local neighborhood in the previous layer, thus it can be seen as a local feature detector. The outputs of the units in the same position in different feature maps can be assumed as a feature vector of the same area. Thus, as CNNs take 2-D image as its entry, it's indispensable to produce 2-D images from grouped character hypotheses. First, adjacent points are connected. Then, strokes are extended to the width of 3 pixels. Finally, anti-aliasing and resizing the image to 32 $\times$ 32 are carried out. Rejection to results of character recognition is certainly important in handwritten recognition, for many samples are quite irregular, unconstrained, or illegal. In paper [8], the outputs of LeNet-5 are set with error-correcting codes (EC codes), thus LeNet-5 has the ability to reject illegal samples. EC codes are also applied in our experiments. When the feed-forward propagation of a sample finishes, the outputs of the CNNs are converted to a vector of EC code. Then, the Hamming distances between this EC code and all standard EC codes of 26 alphabets are calculated respectively. If one of the 26 Hamming distances is greater than a predefined rejection distance, its corresponding recognition result will be rejected. In the end, the left recognition results are sorted by Hamming distances in ascending order and TOPX results are thus obtained [3].

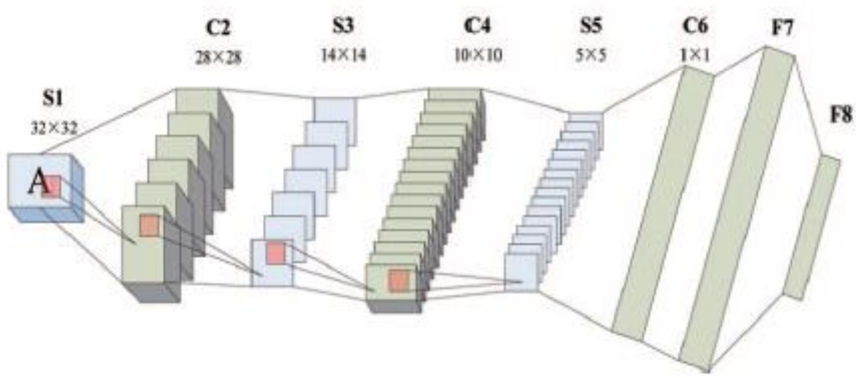


Figure 10. The basic architecture of LeNet-5

Conclusion

Many advances and changes are achieved in the field of handwriting text recognition, over the last decade. However, Most of proposed system still imperfect. In this review, many approaches used in the recognition systems field are overviewed. In this review, Different Pre-processing, segmentation approaches and various classifiers are also discussed.

References

[1] J Mantas. **An overview of character recognition methodologies. Pattern Recognition.**

[2] A. L. Koerich, R. Sabourin, C. Y. Suen . **Large vocabulary off-line handwriting recognition: A survey.**

[3] Aiquan Yuan, Gang Bai, Po Yang, Yanni Guo, Xinting Zhao College of Information Technical Science Nankai University **. Handwritten English Word Recognition based on Convolutional Neural Networks.**

[4] Patrick Doetsch, Michal Kozielski and Hermann Ney. **Fast and robust training of recurrent neural networks for offline handwriting recognition.**

[5] Rejean Plamondon, Fellow, IEEE and Sargur N. Srihari, Fellow, IEEE **;On-Line and Off-Line Handwriting Recognition: A Comprehensive Survey.**

[6] Wojciech Kacalak Koszalin University of Technology . **HANDWRITING RECOGNITION METHODS USING ARTIFICIAL NEURAL NETWORKS.**

[7] Banumathi. K. L . **Line and word Segmentation of Kannada HandwrittennText documents using Projection Profile Technique.**

[8] Richa Goswami . **A Review on Character Recognition Techniques.**

[**9**] Kakade Snehal Satwashi, Prof.Dr.V.R.Pawar. Department of Electronics and Telecomunication University of Savitribai Phyle,India**. English Text Localisation and Recognition from Natural Secne Image.**

[**10**] Boris Epshtein, Eyal Ofek, and Yonatan Wexler. **Detecting text in natural scenes with stroke width transform**.

[11] Jesper Dürebrandt . **Segmentation and Beautification of Handwriting using Mobile Devices.**

[12] Palaiahnakote Shivakumara, b Anjan Dutta, c Umapada Pal and a Chew Lim Tan. School of Computing, National University of Singapore, Singapore. **A New Method for Handwritten Scene Text Detection in Video.**

[13] Olivier Morillot, Laurence Likforman-Sulem, and Emmanu`ele Grosicki. New baseline correction algorithm for text-line recognition with bidirectional recurrent neural networks.

[14] Salvador Espana-Boquera, Maria Jose Castro-Bleda, Jorge Gorbe-Moya, and Francisco Zamora-Martinez. **Improving offline handwritten text recognition with hybrid hmm/ann models**.

[15] Batuhan Balci. Dan Saadati. Dan Shiferaw. Stanford university**. Handwritten Text Recognition using Deep Learning**

[16] Giuseppe AiròFarulla , Nadir Murru , Rosaria Rossini **.A fuzzy approach to segment touching characters**

[17] Amit Choudharya, Rahul Rishib, Savita Ahlawatc. **A New Character Segmentation Approach for Off-Line Cursive Handwritten Words**

[18] Myat Lay Phyu, Kiyota Hashimoto. **Burmese Word Segmentation with Character Clustering and CRFs.**

[19] Yi Ku and M.Shridhar. **Character segmentation in handwriting words-An overview.**

[20] E. Kavallieratou, E. Stamatatos, N. Fakotakis, and G. Kokkinakis. **Handwritten Character Segmentation Using Transformation-Based Learning.**

[21] Magdalena Brodowska**. Oversegmentation methods for character segmentation in off-line cursive handwritten word recognition – An overview**

[22] Wojciech Kacalak. **HANDWRITING RECOGNITION METHODS USING ARTIFICIAL NEURAL NETWORKS**

[23] Yusuf Perwej, Ashish Chaturvedi **.Neural Networks for Handwritten English Alphabet Recognition**