# Realtime Character Translator

**Girish B. Baviskar[1], P.M.Mahajan[2]**

[1,2] Department of Electronics & Telecommunication Engineering, J.T.Mahajan College of Engineering, Faizpur, India

**Abstract:** ***Real time character translator project is one of the innovating application to character recognition. Artificial neural networks have been effectively applied to document analysis and recognition of isolated handwritten and printed characters with widely recognized successful results. However, many other document processing tasks like pre-processing, layout analysis, character segmentation, word recognition, and signature verification have been effectively faced with very promising results. It is being advantages to have a converter that can convert English into many other languages so it is being good to improve understanding of many concepts.. It has numerous applications that includes, reading aid for blind, bank cheques and conversion of any hand written document into structural text form. Neural Network (NN) with its inherent learning ability offers promising solutions for handwritten character recognition. This project identifies the most suitable NN for the design of hand written English character recognition system and after recognition of sentences or word it can convert that sentences to any other language selected. This project focus on English character recognition using feed forward neural network. We emphasis on capital, small and special symbols of English language. After recognizing characters we translate the paragraph into Hindi language and many more languages in world by using GOOGLE transliteration.***

**Keywords:** Neural Network, OCR, MLP, Feed forward back propagation, SVM, ANN

## 1. Introduction

Artificial neural networks have been extensively applied to document analysis and recognition. Most efforts have been devoted to the recognition of isolated handwritten and printed characters with widely recognized successful results. However, many other document processing tasks like pre-processing, layout analysis, character segmentation, word recognition, and signature verification have been effectively faced with very promising results. Handwriting recognition has been one of the active and challenging research areas in the field of image processing and pattern recognition. It has numerous applications that includes, reading aid for blind, bank cheques and conversion of any hand written document into structural text form. Neural Network (NN) with its inherent learning ability offers promising solutions for handwritten character recognition. This project identifies the most suitable NN for the design of hand written English character recognition system. This project focus on English character recognition using feed forward neural network. We emphasis on capital, small and special symbols of English language. After recognizing characters we translate the paragraph into Hindi language by using GOOGLE transliteration. This project can be used in banks to read cheques written in English language.

Optical Character Recognition (OCR) is a process by which text characters can be input to a computer by providing the computer with an image. The computer uses an OCR Engine--a computer program with the specific function of making a guess which letter (recognizable to a computer) an image (recognizable to a human) represents. Paperless includes an OCR Engine, which it uses to recognize text and numerical values. In order to understand how the OCR Engine in Paperless produces OCR results, it is useful also to understand how OCR Engines make these guesses.

### A. How OCR Engines Work

An OCR engine scans an image for elements that resemble letters it is programmed to recognize.

OCR engines use sets of parameters to discern one character from another. For example:

- The letters E and F look a lot alike; the most-noticeable differences between the two characters is the horizontal bar at the bottom of the letter E.
- The letters P and D look a lot alike. There are two primary differences between the letters: the letter P has a vertical line that extends beyond the loop shape at the top of the letter; the letter D does not.
- The letters e and a look a lot alike.
- The letters q and the number 9 look a lot alike in certain fonts.
- The number 2 and the letter Z look a lot alike.
- The semicolon (;) and the colon (:) symbols are nearly-identical.
- The period (.) and the comma (,) symbols are nearly-identical.

At the most-basic level, OCR locates points in a text that resemble characters it is trained to recognize; once it finds what it believes is a match, it returns a letter (recognizable to the computer) to OCR results.This also limits the ability of OCR in some cases; the Roman letter B and the Greek capital Beta symbol (ß) also look a lot alike. B is on the standard UTF-8 character Map; ß is not. If a character is not on an OCR engine's list of characters to recognize, it will either not recognize the character or interpret it as something that is on its list. Similarly, the OCR engine needs to make decisions that separate the letter capital letter k: K from the combination of a pipe character with the less-than symbol: Both look very similar visually, but to a human reader, they may represent very-different things.OCR engines are also programmed to recognize certain fonts. Thus, if the OCR engine is programmed to recognize Bitstream Vera Sans, it will recognize text characters rendered in Bitstream Vera Sans. If the OCR engine has been instructed to recognize Bitstream Vera Sans, but it has not been configured to recognize Linux Libertine, it may not recognize text in Linux Libertine with the same (or any) degree of accuracy compared to what it is able to recognize Bitstream Vera Sans with.

### B. The OCR Process

1. An image of the document is acquired by the computer.
2. The image is submitted as input to an OCR engine.
3. The OCR engine matches portions of the image to shapes it is instructed to recognize.
4. Given logic parameters that the OCR engine has been instructed to use, the OCR engine will make its best guess as to which letter a shape represents.
5. OCR Results are returned as text.

### C. How Does OCR Work?

There are two basic methods used for OCR: Matrix matching and feature extraction. Of the two ways to recognize characters, matrix matching is the simpler and more common. m Matrix Matching compares what the OCR scanner sees as a character with a library of character matrices or templates. When an image matches ,one of these prescribed matrices of dots within a given level of similarity, the computer labels that image as the corresponding ASCII character.Feature Extraction is OCR without strict matching to prescribed templates. Also known as Intelligent Character Recognition (ICR), or Topological Feature Analysis, this method varies by how much "computer intelligence" is applied by the manufacturer. The computer looks for general features such as open areas, closed shapes, diagonal lines, line intersections, etc. This method is much more versatile than matrix matching.Matrix matching works best when the OCR encounters a limited repertoire of type styles, with little or no variation within each style. Where the characters are less predictable, feature, or topographical analysis is superior.

## 2. Related Work

Today's' computers equipped with cameras or optical scanners can read documents and provide their faithful electronic reproduction. In spite of these technological achievements, however, stacks of documents still flood desks of most offices. Whereas documents can be read and accurately stored, the processing required for extracting information is still only in its infancy. Unfortunately, either the presence of noise or the hardly predictable document structure makes it very hard to extract information automatically. In the last decade, the decreasing cost of document acquisition, storage and processing, and the renewal of interest in Artificial Neural Networks (ANNs) have given rise to many novel solutions to different tasks of document processing. The topic of Document Image Analysis and Recognition (DIAR) is thoroughly dealt with in books and survey papers (see e.g. [1, 2, and 3]. In spite of the emphasis on tasks like OCR and word recognition, the application of ANNs to other important DIAR tasks has not received much attention yet. An extensive bibliography on applications of artificial neural networks to document processing tasks1 shows that most papers deal with OCR-related tasks (65%) and with word recognition (15%). To the best of our knowledge, important complementary topics have not been the subject of surveys like for OCR and word recognition [4, 5]. The purpose of this paper is to fill this gap by providing an introduction to most significant problems of DIAR where connectionist-based approaches have demonstrated their effectiveness. To narrow the scope of the paper, we focused attention on applications dealing with document images only, thus leaving out the large literature related to the on-line processing of cursive words and signatures. To face document analysis and recognition tasks one needs to select adequate neural architectures and learning schemes, and to conceive appropriate representations of the data to be processed. Most of the applications to DIAR rely on traditional schemes based on multilayer perceptions (MLPs, e.g. [6]) and do not propose novel methodologies. The original contributions concern the way MLPs are applied to the specific task.

There are a number of straightforward applications to tasks like filtering and noise removal, character, word, and graphical item recognition. The massive experimentation carried out in the last few years demonstrates that, unfortunately, in spite of very interesting and promising results, critical issues of learn ability and computational capabilities often arise when dealing with real-world problems. The pattern representation plays also a very crucial role for the effectiveness of the proposed solution. In many tasks, the input to be pre-processed has typically a flat representation based on a vector of features. For instance, zoning (based on grouping together features in each region of a grid superimposed to the character) is frequently used in OCR for feeding a neural network with a sub-sampled image of variable-size characters. Whereas a flat representation is appropriate for many tasks, structural representations seem to be very appropriate for all tasks which involve the whole document or any parts which exhibit relevant structure. The recent developments in the field of learning in structured domains (see e.g. [7, 8]) offer new unexplored and promising research directions, some of which are reviewed in the following.

This paper is organized in four sections. First section discusses feature extraction and classification. Second section discussion proposed method. Third section discusses results and discussion and last section describes conclusion and future scope.

## 3. Feature Extraction and Classification

Handwritten Recognition refers to the process of translating images of hand-written, typewritten, or printed digits into a format understood by user for the purpose of editing, indexing/searching, and a reduction in storage size. Handwritten recognition system is having its own importance and it is adoptable in various fields such as online handwriting recognition on computer tablets, recognize zip codes on mail for postal mail sorting, processing bank check amounts, numeric entries in forms filled up by hand and so on. There are two distinct handwriting recognition domains; online and offline, which are differentiated by the nature of their input signals. In offline system, static representation of a digitized document is used in applications such as cheque, form, mail or document processing. On the other hand, online handwriting recognition (OHR) systems rely on information acquired during the production of the handwriting. They require specific equipment that allows the capture of the trajectory of the writing tool. Mobile communication systems such as Personal Digital Assistant (PDA), electronic pad and smart-phone have online handwriting recognition interface integrated in them. Therefore, it is important to further improve on the recognition performances for these applications while trying to constrain space for parameter storage and improving processing speed.Many current systems use Discrete Hidden Markov Model based recognizer or a hybrid of Neural Network (NN) and Hidden Markov Model (HMM) for the recognition. Online information captured by the input device first needs to go through some filtration, preprocessing and normalization processes.

After normalization, the writing is usually segmented into basic units (normally character or part of character) and each segment is classified and labeled. Using HMM search algorithm in the context of a language model, the most likely word path is then returned to the user as the intended string. Segmentation process can be performed in various ways. However, observation probability for each segment is normally obtained by using a neural network (NN) and a Hidden Markov Model (HMM) estimates the probabilities of transitions within a resulting word path. This research aims to investigate the usage of support vector machines (SVM) in place of NN in a hybrid SVM/HMM recognition system. The main objective is to further improve the recognition rate by using support vector machine (SVM) at the segment classification level. This is motivated by successful earlier work by Ganapathiraju in a hybrid SVM/HMM speech recognition (SR) system and the work by Bahlmann in OHR. Ganapathiraju obtained better recognition rate Compared to hybrid NN/HMM SR system. In this work, SVM is first developed and used to train an OCR system using character databases. SVM with probabilistic output are then developed for use in the hybrid system. Eventually, the SVM will be integrated with the HMM module for word recognition. Preliminary results of using SVM for character recognition are given and compared with results using NN reported by Poisson. The following databases were used: IRONOFF, UNIPEN and the mixture IRONOFF-UNIPEN databases.

A character recognition system basically deals with the recognizing offline handwritten character. Typically it can be classified as the following two types[8].

- Online recognition and
- Offline recognition

- **Online Character Recognition**

In case of online character recognition, there is real time recognition of characters[6]. Online systems have better information for doing recognition since they have timing information and since they avoid the initial search step of locating the character as in the case of their offline counterpart. Online systems obtain the position of the pen as a function of time directly from the interface. Offline recognition of characters is known as a challenging problem because of the complex character shapes and great variation of character symbols written in different modes.

- **Offline Character Recognition**

In case of offline character recognition, the typewritten/ handwritten character is typically scanned in form of a paper document and made available in the form of a binary or gray scale image to the recognition algorithm. Offline character recognition is a more challenging and difficult task as there is no control over the medium and instrument used [7]. The artifacts of the complex interaction between the instrument medium and subsequent operations such as scanning and binarization present additional challenges to the algorithm for the offline character recognition. Therefore offline character recognition is considered as a more challenging task then its online counterpart.

**A. Different Phases of the Character Recognition System**

The Pre-processing step aims to improve the image data or the image features that required for the further processing. The pre-processing is a series of operations performed on the scanned input image. It essentially enhances the image. It involves converting an input image to binary image, noise removing, dilation operation, line segmentation and digit segmentation and normalization.

Feature Extraction is a very important step for any character recognition system. This step involves the procedures like shape information or style which is very much useful for the classification of the pattern. The feature extraction stage analyses a text segment and selects a set of features that can be used to uniquely identify the text segment. Classification stage uses the features extracted to identify the text segment according to the algorithm. The task is to compare the testing patterns and minimizing the error rate and correct classification of the pattern.Post-processing involves various approaches dictionary lookup and statistical approach or neural network recognition for the correct recognition.

**B. K-nearest Neighbor classifier**

K-nearest neighbor classifier approach has been used for the Gujarati character recognition. K-nearest neighbor classifier has been found very good results for the English characters. It used the k-nearest samples to test sample and identifies it to that class which has the largest number of votes. The nearest neighbor is found by using the Euclidean distance measure. For 1-NN classifier the best recognition rate achieved was 67% in the binary feature space and in regular moment space the rate was 48%.

**C. The Minimum Hamming Distance Classifier**

This approach has been used for the Gujarati character recognition. The Minimum Hamming Distance Classifier uses the Hamming Distance between the sample and the class centroids built using the training sets to classify characters. It is assumed that the image pixels have a Bernoulli distribution. Then the hamming distance is the sum of the absolute pixel difference (in binary space) between the class centroids and the image of the character being classified. Using this approach the recognition rate was only 39%.

**D. Feed forward back propagation neural network classifier**

In a feed forward back propagation neural network is proposed for the classification of the Gujarati numerals.

Various techniques are used in the preprocessing phase before implementing classification of numerals. Gujarati numerals are based on very sharp curves and curves are irregular, to handle this situation here in this work, various profiles of digits are used as template to identify various digits. In this very simple but effective, feature extraction technique the use of four different profiles, horizontal, vertical, and two diagonals, is suggested. The overall performance of this proposed network is as high as 81.66%.
A handwritten character recognition system using multilayer Feed forward neural network is proposed in [13]. Three different orientations, namely, horizontal, vertical and diagonal directions are used for extracting 54 features from each character. In addition, 9 and 6 features are obtained by averaging the values placed in zones row wise and column wise, respectively. As a result; every character is represented by 69, that is, 54 +15 features. From the test results it is identified that the diagonal method of feature extraction yields the highest recognition accuracy of 98% for 54 features and 99% for 69features.

**E. KNN and PCA classifier**

In [15] they are using KNN classifier and PCA (to reduce dimensions of feature space) and used Euclidean similarity measure to classify the numerals. KNN classifier yielded 90 % as recognition rate whereas PCA scored recognition rate of 84%. The comparison of KNN and PCA is made and it can be seen that KNN classifier has shown better results as compared to PCA classifier.

**F. SVM Classifier**

In [14] authors propose the Support Vector Machine (SVM) based recognition scheme towards the recognition of Gujarati handwritten numerals. A technique based on affine invariant moments for feature extraction is applied and the recognition rate of 91% approximately.

## 4. Proposed Methodology

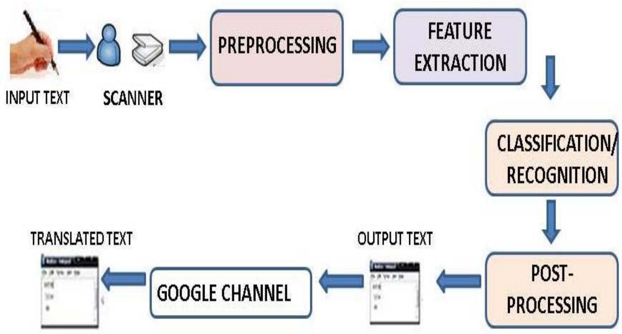


**Figure 1:** System Block Diagram

**A. Image Acquisitions**

In Image acquisition, the recognition system acquires a scanned image as an input image. The image should have a specific format such as JPEG, BMP etc. This image is acquired through a scanner, digital camera or any other suitable digital input device. Data samples for the experiment have been collected from different individuals.

**B. Pre-processing**

The main steps involved are:

- **Binarization**

Image binarization converts an image of up to 256 gray levels to a black and white image. Frequently, binarization is used as a pre-processor before OCR. In fact, most OCR packages on the market work only on bi-level (black & white) images. The simplest way to use image binarization is to choose a threshold value, and classify all pixels with values above this threshold as white, and all other pixels as black. The problem then is how to select the correct threshold. In many cases, finding one threshold compatible to the entire image is very difficult, and in many cases even impossible. Therefore, adaptive image binarization is needed where an optimal threshold is chosen for each image area. Binarization is done using thresholding with a manually defined threshold or Otsu's method.

- **Noise removal**

Noise reduction is the process of removing noise from a signal. Noise reduction techniques are conceptually very similar regardless of the signal being processed, however a priori knowledge of the characteristics of an expected signal can mean the implementations of these techniques vary greatly depending on the type of signal. All recording devices, both analog anddigital, have traits which make them susceptible to noise. Noise can be random or white noise with no coherence, or coherent noise introduced by the device's mechanism or processing algorithms. Images taken with both digital cameras and conventional film cameras will pick up noise from a variety of sources. Many further uses of these images require that the noise will be (partially) removed - for aesthetic purposes as in artistic work or marketing, or for practical purposes such as computer vision.

In salt and pepper noise (sparse light and dark disturbances), pixels in the image are very different in color or intensity from their surrounding pixels; the defining characteristic is that the value of a noisy pixel bears no relation to the color of surrounding pixels. Generally this type of noise will only affect a small number of image pixels. When viewed, the image contains dark and white dots, hence the term salt and pepper noise. Typical sources include flecks of dust inside the camera and overheated or faulty CCD elements.

In Gaussian noise, each pixel in the image will be changed from its original value by a (usually) small amount. A histogram, a plot of the amount of distortion of a pixel value against the frequency with which it occurs, shows a normal distribution of noise. While other distributions are possible, the Gaussian (normal) distribution is usually a good model, due to the central limit theorem that says that the sum of different noises tends to approach a Gaussian distribution. In either case, the noises at different pixels can be either correlated or uncorrelated; in many cases, noise values at different pixels are modeled as being independent and identically distributed, and hence uncorrelated.

- **Tradeoffs**

In selecting a noise reduction algorithm, one must weigh several factors such as the available computer power and time available: a digital camera must apply noise reduction in a fraction of a second using a tiny onboard CPU, while a desktop computer has much more power and time, whether sacrificing some real detail is acceptable if it allows more noise to be removed (how aggressively to decide whether variations in the image are noise or not), the characteristics of the noise and the detail in the image, to better make those decisions.

### C. Segmentation

Character segmentation seeks to decompose a sequence of characters into individual symbols. Segmentation strategies can be divided into three main categories [3]. Dissection methods partition the input image into sub-images having “character-like” properties. Recognition based methods rely on the integration of segmentation and recognition. Holistic approaches avoid segmentation by recognizing entire words as units. In this section we analyze dissectionmethods employing ANNs.

### D. Neural Network based Classifier

Neural Network (NN) techniques offer a promising solution as classifiers in the handwritten character recognition system. The image after resizing is taken as an input. The classification capability of the network depends on the architecture and learning rule. The architectures considered in this paper are feed forward architecture, nearest neighborhood and radial basis function architecture. To evaluate the performance of the proposed method the handwritten uppercase English alphabets were collected from different individual writers. 560 samples were used for training purpose. Around 10 images (containing paragraphs) were used for testing. The proposed recognition system has been implemented using JAVA. The recognition systems were designed using different methods as listed below.

### E. Feed Forward Back Propagation Neural Network classifier

The scanned image is taken as dataset/ input and feed forward architecture is used. As each image is resized into 30X20 pixels, the input layer has 600 neurons equal to the total number of pixels. The number of output neurons is based on the number of alphabets. As all the English alphabets are used, the output layer has 26 neurons. All the neurons use log-sigmoid transfer functions. The back propagation algorithm with momentum and adaptive learning rate is used to obtain the parameters of the network. Two Hundred different handwritten data sets were used for training the neural network. The number of hidden layers and the number of neurons in each layer are to be obtained through trial and error. Through numerous simulations it was identified that a maximum of two hidden layers and a maximum of 100 neurons in each hidden layer would be sufficient for character recognition. Further increase in the number of neurons did not considerably improve the accuracy. This feed forward neural network architecture was trained for a target MSE of 10e-8. After the network is satisfactorily trained, the parameters of the trained network are fixed to enable testing and the network training parameters are shown in Table 1. The results obtained are shown in Table. 1.

**Table 1**: Feed forward Neural Network parameters:

| Feedforward Neural Network parameters | |
|---|---|
| Input nodes | 600 |
| Hidden layers | 2 |
| Hidden layers nodes | 100 each |
| Output nodes | 26(Capital letters)+26(Small letters)+10 (Digits0-9 ) + 18(Special Symbols) |
| Training algorithm | Gradient descent with momentum training and adaptive learning |
| Performance function | Mean Square Error (MSE) |
| Training goal achieved | 10e-8 |

Number of training samples: 560
Number of testing samples: 450
Feed Forward Neural Network:
Number of correctly recognized characters: 427
Total Number of characters: 450

**Recognition rate: 95%**

The output of the $i^{th}$ layer is defined by

$$a^i = \log sig(w^i a^{i-1} + b^i)$$

Where,

$a^o = P$ & $i = [1,2,3]$

$w^i$= Weight vector of $i^{th}$ layer

$a^i$ = Output of $i^{th}$ layer

$b^i$ = Bias vector for $i^{th}$ layer

P= Input vector for network

### *E. Translation of Character*

Respective recognized character after completion of post processing further characters prepares some sentences those are input to “GOOGLE TRANSLATE” is converted into any other language as per need.

## 5. Results and Discussion

Feed forward neural network is used for English language character recognition. We have tested our experiment on capital letters, small letters digits and special symbols. 560 samples were used for training of neural network. Out of 560 samples, 182 samples of capital letters, 182 samples of small letters, 70 samples were of digits and remaining samples of special symbols. Then testing is performed on around 10 images (containing paragraphs) and having 450 characters. Our experiment has detected 427 characters correctly hence our proposed method have accuracy rate of 95%. This project is implemented in JAVA.

## 6. Conclusion and Future Scope

An off-line handwritten character recognition system with Feed forward NN for recognizing handwritten English characters has been described in this project. The feature extraction and classification tasks are performed together as a single process in the proposed system unlike in typical handwritten recognition systems in which these tasks are carried out in two different stages. As a result, the proposed system is found to be less complex and allows faster recognition of characters. The proposed classifier have been trained with 560 sets of data and extensively tested. Experimental results show that the feed forward neural network is distinctly superior to the other classifiers in recognizing the handwritten English alphabets because recognition rate is 95%. Further investigation was carried out to identify the recognition rates for each letter of alphabet. This would help to estimate the recognition rate irrespective of the handwritten content. It was identified that the Feed forward NN outperformed the remaining classifiers. The proposed system will find useful applications in recognizing the handwritten names, reading documents and conversion of any handwritten document into structural text form.

Further improvements may be possible with a more complex Feedforward NN architecture but this would also increase the computation complexity. Therefore, combination of a standard feature extraction technique with Feedforward NN may provide better solutions. Further the work may be carried out for cursive handwritten letters.

## References

[1] L. O'Gorman and R. Kasturi, *Document Image Analysis*. Los Alamitos, California:IEEE Computer Society Press, 1995.
[2] Nagy, "Twenty years of document image analysis in PAMI," *IEEE TPAMI*, vol. 22, no. 1, pp. 38–62, 2000.
[3] R. G. Casey and E. Lecolinet, "A survey of methods and strategies in character segmentation," *IEEE TPAMI*, vol. 18, no. 7, pp. 690–706, 1996.
[4] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
[5] M. D. Garris, C. L. Wilson, and J. L. Blue, "Neural network-based systems for handprint OCR applications," *IEEE Trans. Image Processing*, vol. 7, no. 8, pp. 1097–1112, 1998.
[6] Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in *Parallel Distributed Processing* (D. E. Rumelhart and J. L. McClelland, eds.), vol. 1, ch. 8, pp. 318–362, Cambridge: MIT Press, 1986.
[7] P. Frasconi, M. Gori, and A. Sperduti, "A general framework for adaptive processing of data structures," *IEEE TNN*, vol. 9, no. 5, pp. 768–786, 1998.
[8] P. Frasconi, M. Gori, and A. Sperduti, "Guest editors' introduction: special section on connectionist models for learning in structured domains," *IEEE Trans. Knowledgeand Data Engineering*, vol. 13, no. 2, pp. 145–147, 2001.
[9] M. Gori, S. Marinai, and G. Soda, "Artificial neural networks for document analysis and recognition," tech. rep., N. 1/2003 - DSI - University of Florence, "http://www.dsi.unifi.it/ANNxDAR/", 2003.
[10] Z. Chi and K.W.Wong, "A two-stage binarization approach for document images," in *Proc. Int'l Symp. Intelligent Multimedia, Video and Speech Processing*, pp. 275–278, 2001.
[11] P. Whichello and H. Yan, "Linking broken character borders with variable sized masks to improve recognition," *Pattern Recognition*, vol. 29, no. 8, pp. 1429–1435, 1996.
[12] P. Stubberud, J. Kanai, and V. Kalluri, "Adaptive image restoration of text images that contain touching or broken characters," in *ICDAR 95*, pp. 778–781, 1995.
[13] P. Martin and C. Bellisant, "Low-level analysis of music drawing images," in *ICDAR91*, pp. 417–425, 1991.
[14] N. Rondel and G. Burel, "Cooperation of multilayer perceptrons for the estimation of skew angle in text document images," in *ICDAR 95*, pp. 1141–1144, 1995.
[15] R. Palaniappan, P. Raveendran, and S. Omatu, "New invariant moments for nonuniformly scaled images," *PAA*, vol. 3, no. 2, pp. 78–87, 2000.