Spectrum of Engineering Sciences
ISSN (e) 3007-3138 (p) 3007-312X Volume 3, Issue 6, 2025

AN EFFICIENT OFF-LINE HANDWRITTEN ENGLISH ALPHABET CHARACTER RECOGNITION BASED ON HIDDEN MARKOV MODEL AND DISCRETE WAVELET TRANSFORM

Nimra Asif[*1], Imran Tauqir[2], Adil-Masood Siddiqui[3]

[*1,2,3]*Dept. of Electrical Engineering, National University of Science and Technology, Islamabad, Pakistan*

[*1]nimraasif51@gmail.com, [2]imrantqr.dradil}@mcs.edu.pk

DOI: https://doi.org/10.5281/zenodo.15637750

Keywords
Hidden Markov Model (HMM), Character Recognition, Geometric based feature extraction, Computational efficiency.

Article History
Received on 03 May 2025
Accepted on 03 June 2025
Published on 11 June 2025

Corresponding Author: *
Nimra Asif

Abstract
Computational efficiency is a matter of great concern in state-of-the-art English alphabet character recognition systems. In this paper, nine state Hidden Markov Model (HMM) for character recognition has been presented. Alphabetical character images are being divided into nine blocks that corresponds to nine respective states of HMM. Corresponding local features of the character are being extracted by using geometric based feature extraction algorithm. Training of the HMM is done by means of the Baum-Welch algorithm. Computational cost of proposed model is minimized by employing Discrete Wavelet Transform (DWT) prior to other dimensionality reduction techniques. The recognition is performed using a Viterbi algorithm to perform best path search in combinations of various character models. Experimental results on handwritten English alphabet character databases demonstrate that recognition accuracy of proposed model is comparable to the existing techniques with reduced computational cost.

INTRODUCTION

Classification of English alphabet character images under un-controlled environment require a robust algorithm. However, computational cost of these algorithms is a serious challenge. Feature extraction, feature selection and classification are the three major steps involved in Automatic Character Recognition (ACR). Statistical, Global transformation and series expansion and Geometric and topological features are the three main classifications of the feature extraction methods [1].

Comparing character images in their original image dimensions is computationally expensive. Therefore, it is necessary to reduce the image dimensions using transformation approach that retains significant image features. Statistical, Global transformation and series expansion, and Geometric and topological features are the three main classifications of the feature extraction methods [1]. The statistical features represent the character image as statistical distribution of points. Zoning, Crossing and Distances, and Projections are the various methods used for statistical feature extraction. However, if we have a limited amount of information, these methods are inadequate to solve a more general solution. Global transformation and series expansion include various techniques such as Fourier transform,

Wavelets, Moments and Karhunen-Loeve Expansion. In this method, features provide sound representation and sound normalization of shapes however, high frequency information is required for more complex character shapes. Structural features are based on geometrical and topological properties of the character. Loops, curves, lines, T-point, cross, aspect ratio, strokes and their directions and inflection between two points are used as structural features. In this method, relationship between the components in the shape is highly expressed.

Besides feature extraction and feature selection methods, classifiers also have significant impact on the performance of Handwritten Character Recognition (HCR) system. Numerous classification models have been developed such as Support Vector Machine (SVM) [2], Minimum Distance Classifier [3], Neural Networks (NN) [4] and HMM [5]. SVM has been used in binary classification of data points. RASHID et al. [11] used NN for English alphabet character recognition in combination with HMM.

Most of the aforementioned handwritten character recognition techniques focus on improving the recognition rate and paying a little attention to reduce the computational complexity. This paper focuses on efficient HCR system using local geometric properties of the character skeleton for feature extraction and nine state HMM for classification. The proposed model employs DWT, Universe of Discourse (UD) and Singular Value Decomposition (SVD) techniques that not only reduce image redundancies while retaining informative features but also decrease feature vector length that improves the efficiency of Baum-Welch algorithm and Viterbi algorithm.

The rest of this paper is organized as follows: section 2 presents the existing character recognition methods. Section 3 presents the overall system and describe in detail the proposed feature extraction and classification method along with complexity analysis of the proposed system. Section 4 confer the experimental results. Section 5 presents the comparative study followed by conclusion in section 6.

Literature Review

Character recognition techniques transform high dimensional images into low dimensional feature vectors to improve efficiency. Computational cost of an algorithm depends upon observation vector length and classification model used. PATEL et al. [3] used DWT as frequency domain features. Query image was classified based on the minimum distance between observation vectors of query and database images.

BECERIKLI et al. [7] proposed a hybrid feature extraction approach that was based on the fusion of Local Binary Pattern (LBP) and horizontal vertical projection features. Back Propagation Neural Network (BPNN) was exploited to recognize the test images. However, training process of BPNN is computationally expensive [8]. Majority voting scheme was employed to classify character images.

The number of states have a major impact on the efficiency of HMM based HCR. KIM et al. [9] used directional, mesh and crossing point features as hybrid features and three NN as combined classifiers to classify numeral character images. ALI et al. [10] author presented DCT based HMM to efficiently recognize the high dimensional images. RASHID et al. [11] used two classifiers; NN and HMM. JENA et al. [13] utilized HMM to recognize the Odia character and numbers.

The Proposed Methodology

In this paper we propose a nine state HMM to replace higher order HMMs for classification. The proposed framework is shown in fig. 1. Nine states of HMM, S1, S2, S3, S4, S5, S6, S7, S8 and S9 corresponds to zonal regions/blocks, Z11, Z12, Z13, Z21, Z22, Z23, Z31, Z32 and Z33, obtained by dividing character images into nine regions respectively as shown in figs. 2 and 3. The image dimensions reduced using Haar wavelet. Pre-processing operations (binarization and skeletonization) are performed on character images to make the system reliable and efficient.

The former operation is used to convert gray scale image into a binary image and the later operation is used to reduce the width of the character from numerous pixels to only pixel. Median filter is used to eliminate the noise effect caused by the sensor and circuitry of a scanner or digital camera. Furthermore, the entire skeleton of character is enveloped in a shortest matrix by applying the process of universe of discourse as shown in fig. 4.

Fig. 1. Proposed methodology for Character Recognition.

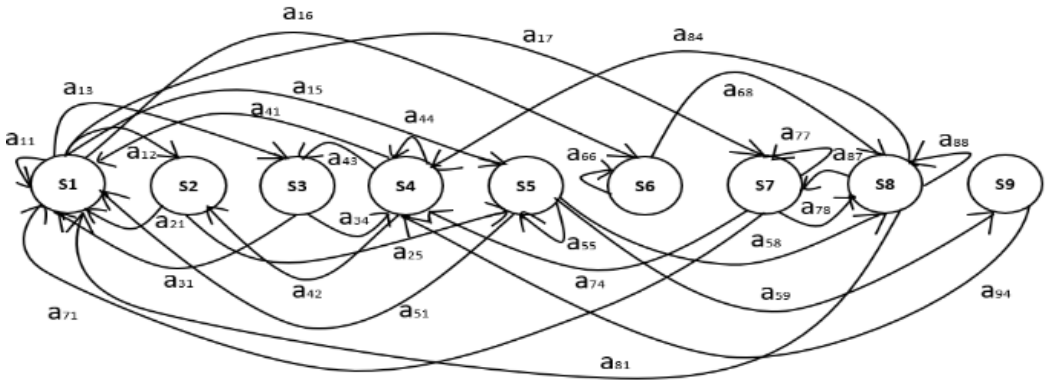


Fig. 2. Nine state HMM for Alphabetical character image.

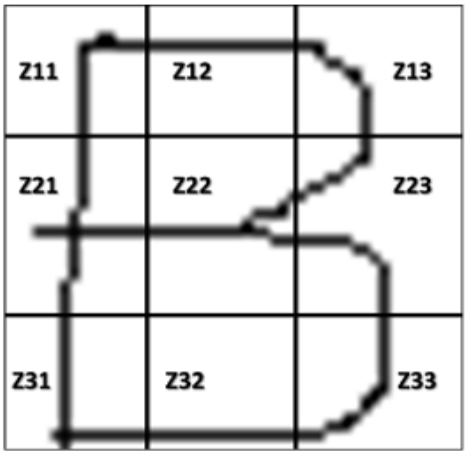


Fig. 3. Character Regions.

$$I = MUN^T \qquad ((1)$$

Where I represent, feature matrix obtained after dimensionality reduction, U is the diagonal matrix that contains singular values of character features, M (mxm) and N (mxm) are orthogonal matrices ($M^T = M^{-1}$, $N^T = N^{-1}$) containing orthonormal arbitrary vectors (Mn) that obeys (2).

$$m_P^T m_q = n_P^T n_q = \sigma_{pq} = \begin{cases} 1, & if\ p = q \\ 0, & if\ p \neq q \end{cases} \qquad ((2)$$

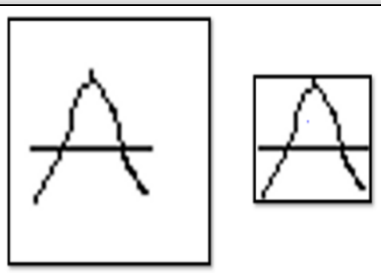


Fig . 4. Character image before and after applying Universe of Discourse.

After the universe of discourse is selected the Zoning technique is used that splits this skeleton of alphabetical character into nine zones/blocks which corresponds to the nine respective states of HMM. The geometric based feature extraction technique is applied on each zone/block one by one to extract local features [1]. These feature coefficients are the visible symbols of model. SVD is used to reduce the feature vector length after feature extraction step using (1). The peculiarity of SVD is that it can be performed on any arbitrary (ixj) matrix.

σ is m×n diagonal matrix of singular values with components, σ_{pq} =1, if p ≠ q and σ_{pq} =0, if p=q. The finite features of a character are then obtained by doing the quantization of the images. The quantized images thus obtained are labelled and corresponding observation sequence for the trained images is obtained. Corresponding trained HMM is generated by applying Baum-Welch algorithm on the basis of observation sequences, transition and emission matrices and estimated transition and estimated emission matrices are thus obtained. HMM trained database is obtained by repeating this process for every single trained image. Thereafter, significant features of test images are extracted. Viterbi algorithm then calculates the recognition probability of the tested observation sequence of tested images by comparing each of the test image with the trained HMM database images. The corresponding test image with the highest matching probability thus displayed on the user screen. The strength of proposed approach is to employ geometric based feature extraction to extract local features of the character skeleton.

Feature Extraction

Before working of feature extraction algorithm, definitions of starter pixels, intersection pixels and minor starter pixels are discussed. Starter pixels are those pixels which have only one neighbor. Intersection pixels are those which have more than one true neighbor, true neighbor means pixels in the direct (pixels in the horizontal and vertical direction) and diagonal direction). Intersection pixels are further classified into 3-neighbour and 4-neighbour intersection pixels. An intersection pixel is 3-neighbour only if true neighbours of a particular pixel are not adjacent to one another as shown in fig. 5. An intersection pixel is 4- neighbour only if each and every true neighbours of a particular pixel are not adjacent to one another as shown in fig. 6. The minor starter pixels are those pixels under consideration which have more than two neighbours and they are only identified during character traversal.

The geometric based feature extraction algorithm mainly includes the following steps:

Step 1. Identification of all the starter pixels in a particular zone, and put them in starters pixel list.

Step 2. Identification of all the intersection pixels in a particular zone and put them in intersections pixel list.

Step 3. Pick a starter pixel from starters pixel list and begin traversal of character skeleton in that zone. If during traversal, pixel encounter is an intersection pixel, stop the current line segment there and put all the unvisited neighbors of this intersection pixel in the minor starters list. Else, pixel encounter is minor starter pixel, stop the current line segment there

and put all the unvisited neighbors of this minor starter pixel (if any) in the minor starters pixel list and removes this minor starter pixel from the minor starters pixel list and get updated minor starters pixel list. This process is repeated for all the starter pixels, until the starter pixel list gets empty. In this way, all the line segments with their pixel locations in a particular zone, are identified as shown in fig. 7.

Step 4. For classification of these line segments into horizontal lines, vertical lines, left diagonal lines and right diagonal lines, will get a direction vector using convention as shown in fig. 8. Take pixel locations of the first line segment and place first pixel location of this line segment at pixel location C of convention [1]. Then subtract first pixel location from second pixel location of this line segment, after subtracting, the new pixel location will get that corresponds to a particular number in convention [1]. This particular number corresponds to a direction vector. This process is repeated for all the remaining pixel locations of this line segment until we will get a direction vector for this line segment. In this way, each line segment is represented by a direction vector.

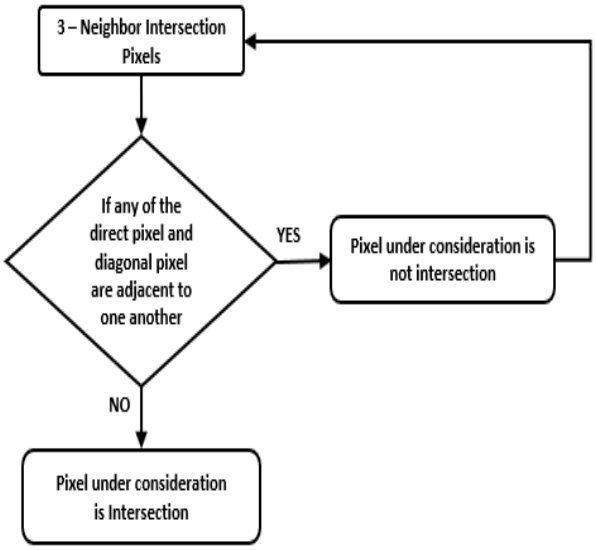


Fig. 5. Flow chart of 3- neighbor intersection pixel.

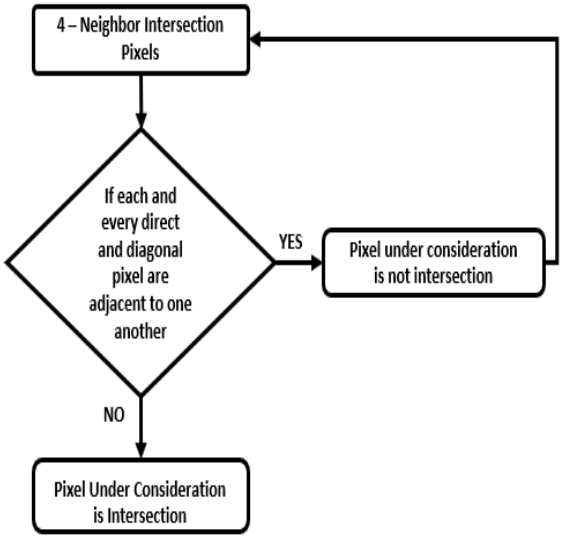


Fig. 6. Flow chart of the 4- neighbor intersection pixel.

Step 5. Now apply the following rules on the direction vector, these rules are:
1. The previous direction was 6 or 2 and the next direction is 8 or 4.
2. The previous direction was 8 or 4 and the next direction is 6 or 2.

If these rules are not applied on the direction vector, will apply the following rules on the direction vector:
1. When the direction type is eight or four, it will be right diagonal line.
If the direction type is six or two, it will be left diagonal line.

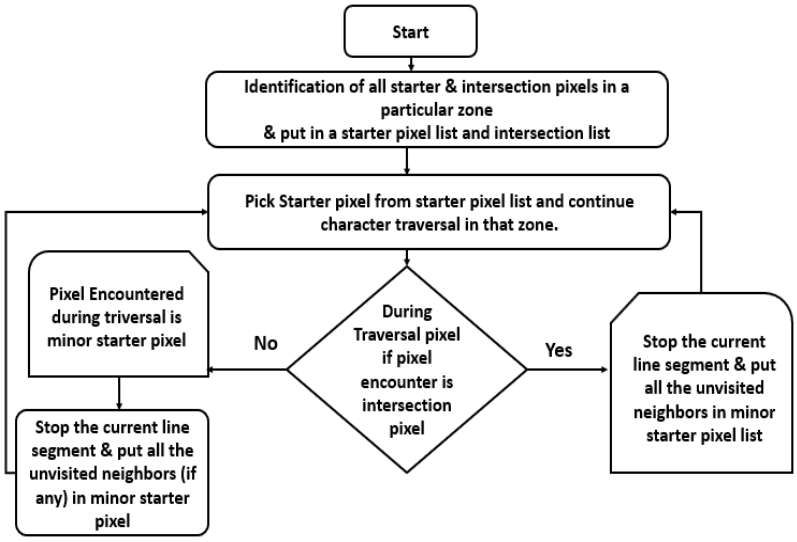
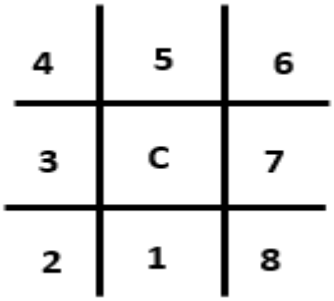


Fig. 7. Flow chart for extracting line segments.



. **Fig. 8.** Naming convention for feature vector.

1. If the direction type is five or one, it will be vertical line.
2. If the direction type is three or seven, it will be horizontal line.

This process is repeated for all the direction vectors as shown in fig. 9. In this way, the line segments are classified as well as their quantity in a particular zone are identified. If we have nine zones/blocks, a feature vector of length nine is generated from each zone/block.

Step 6. Find normalized value and normalized length of these feature values. Using (3) to standardize the number of any certain line form.

$$value = 1 - \left(\frac{number\ of\ lines}{10} \right) * 2 \qquad (3)$$

Using (4) to determine the normalized length of any particular line type,

$$Normalized\ \ length\ of\ a\ specific\ line\ type = 1 - \left(\frac{Total\ \ pixels\ in\ that\ line\ Type}{Total\ \ pixels\ in\ that\ zone} \right) \quad (4)$$

Now, select only those feature coefficients for which classification accuracy of the proposed model is maximum at reduced computational cost. Figure 10 shows singular values of feature matrix of character image as a function of its feature coefficients. It is clear from this figure that singular values corresponding to first five feature coefficients carry most of the information about character image while remaining singular values are negligible. Multiple combinations of these singular values and feature coefficients are tested to improve recognition accuracy. Desired results are obtained when we used first five singular values (σ_H, σ_V, σ_{LD}, σ_{RD}, σ_{NH}) corresponding to number of horizontal lines, number of vertical lines, number of left diagonal lines, number of right diagonal lines and normalized number of horizontal lines. These five feature coefficients are used as features describing image block. These block features are quantized to specific discrete levels in order to reduce computational cost using (5) and (6).

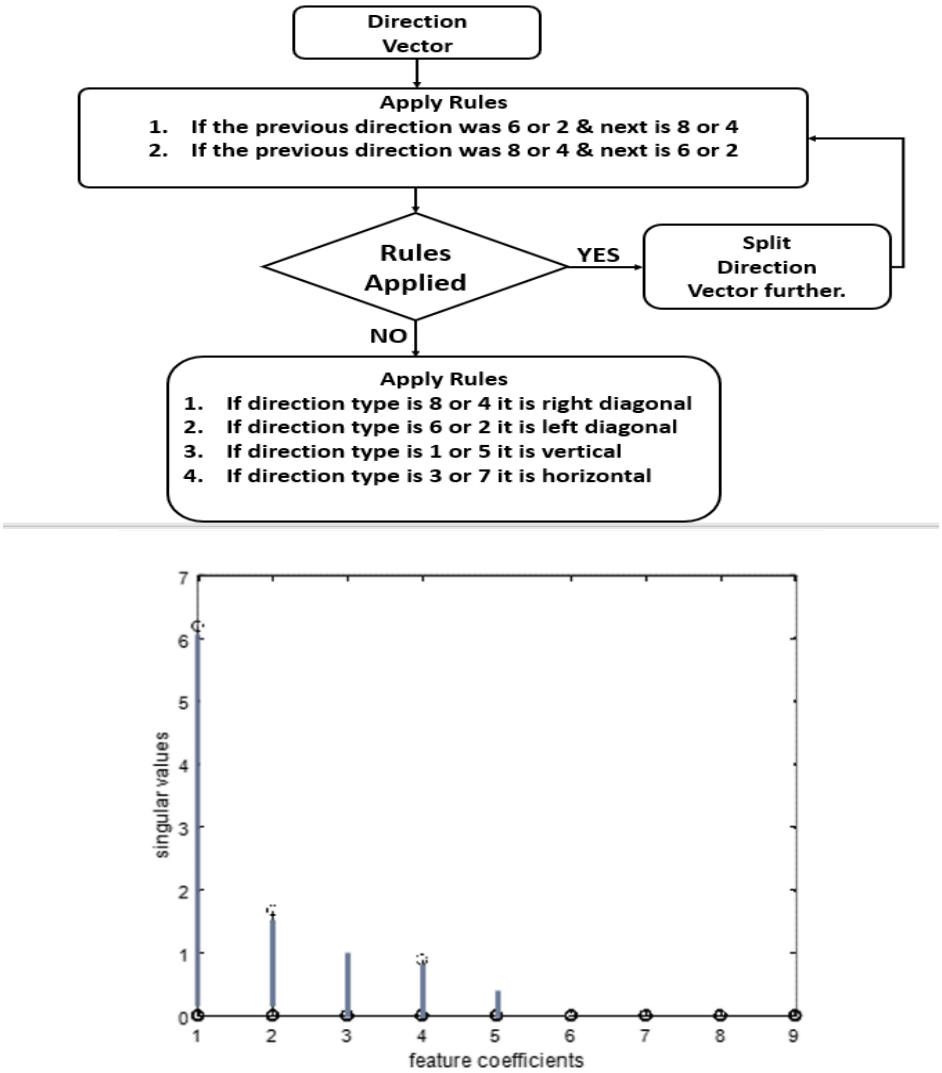


Fig. 9. Flowchart showing the rules applied in the feature vector.

$$Labels = q_n(1)*8 + q_n(2)*60 + q_n(3)*2 \quad q_n(4)*5 + q_n(5)*7 + 6 \quad (7)$$

Where, $q_n(1)$, $q_n(2)$, $q_n(3)$, $q_n(4)$ and $q_n(5)$ represents quantization levels of σ_H, σ_V, σ_{LD}, σ_{RD}, σ_{NH} respectively. Empirically good values of $q_n(1)$, $q_n(2)$, $q_n(3)$, $q_n(4)$ and $q_n(5)$ are found to be 10, 10, 10, 10 and 7 respectively. Thus, for each block there are 45 possible feature values/ labels. An observation sequence is formed for each training image which contains feature labels of all image blocks. This observation sequence is further used to train HMM.

Fig. 10. Singular values for alphabetical character image.

$$f_j^{qn} = \frac{f_j - f_j^{\min}}{\Delta_j} \quad (5)$$

$$\Delta_j = \frac{f_j^{\max} - f_j^{\min}}{L_j} \quad (6)$$

Where $f_j^{\min}$, $f_j^{\max}$ and f_j^{qn} are minimum, maximum and quantized values of feature coefficients respectively. Δ_j Is the difference between two successive quantization levels and L_j is the number of quantization levels. Quantized features are assigned labels using (7).

HMM Training Process

Two major steps are involved in HMM training. In the first step model parameters [A, B, π] are initialized. In proposed model good results are obtained using following initial parameters where:

$$\pi = [100000000]$$

$$B = \frac{1}{N_{sym}} ones\left(N_s, N_{sym}\right) \quad (8)$$

$$N_s = 9,\ N_{sym} = 45$$

Π is initial probability vector, **A** and **B** are transition and emission probability matrices respectively. Figures 11 and 12, show the emission and transition matrices where W_1, W_1, W_2, W_3, W_4, W_5, W_6, W_7, W_8 and W_9 are hidden states of proposed nine states HMM which represents Z11, Z12, Z13, Z21, Z22, Z23, Z31, Z32 and Z33 respectively.

$V_0, V_1, V_2, \ldots, V_{45}$ represents the visible symbols of HMM which are basically the features of alphabet character images extracted using character geometry technique. N_s and N_{sym} represent the number of states of model and number of visible symbols respectively.

In the second phase of HMM training, Baulm-Welch algorithm is used to r-estimate the model parameters. Baulm-Welch algorithm runs in multiple iterations using the observation sequence of training images. In each iteration all possible combinations of hidden states corresponding to observation sequence (b^T) are determined. Probability of transition for a particular visible symbol (b_d) is determined using (9).

$$P\left(S_r \xrightarrow{bd} S_{r+1}\right) = \frac{c\left(S_r \xrightarrow{bd} S_{r+1}\right)}{\sum_{t=1}^{N}\sum_{k=1}^{T} c\left(S_r \xrightarrow{bk} S_l\right)} \quad (9)$$

Where C (·), S_r, S_l and bk represent number of counts of transition, specific hidden state, all possible hidden states and visible symbols respectively. Number of counts is determined using (10).

$$B = \begin{array}{c} \\ W_1 \\ W_2 \\ W_3 \\ W_4 \\ W_5 \\ W_6 \\ W_7 \\ W_8 \\ W_9 \end{array} \begin{array}{c} \begin{array}{cccccc} V_0 & V_1 & V_2 & V_3 & \dots & V_{45} \end{array} \\ \begin{bmatrix} 1/45 & 1/45 & 1/45 & 1/45 & \dots & 1/45 \\ 1/45 & 1/45 & 1/45 & 1/45 & \dots & 1/45 \\ 1/45 & 1/45 & 1/45 & 1/45 & \dots & 1/45 \\ 1/45 & 1/45 & 1/45 & 1/45 & \dots & 1/45 \\ 1/45 & 1/45 & 1/45 & 1/45 & \dots & 1/45 \\ 1/45 & 1/45 & 1/45 & 1/45 & \dots & 1/45 \\ 1/45 & 1/45 & 1/45 & 1/45 & \dots & 1/45 \\ 1/45 & 1/45 & 1/45 & 1/45 & \dots & 1/45 \\ 1/45 & 1/45 & 1/45 & 1/45 & \dots & 1/45 \end{bmatrix} \end{array}$$

Fig. 11. Emission Probability Matrix.

$$C\left(S_r \xrightarrow{bd} S_{r+1}\right) = \sum_{i=1}^{i\max} P\left(\frac{S_i^T}{b^T}\right) * n\left(S_r \xrightarrow{bd} S_{r+1}\right) \quad (10)$$

Where S_i^T and T represents sequence of hidden states and sequence length of visible symbols respectively. After each iteration of Baulm-Welch algorithm model parameters are updated and these parameters are used as initial parameters in next iteration. Updated transition and emission probabilities are determined using (11) and (12) respectively.

$$a_{r,r+1} = \frac{\sum_k P(S_r \xrightarrow{bk} S_{r+1}}{\sum_l \sum_k P\left(S_r \xrightarrow{bk} S_l\right)} \quad (11)$$

$$b_{r+1,d} = \frac{\sum_l P(S_l \xrightarrow{bd} S_{r+1}}{\sum_l \sum_k P\left(S_l \xrightarrow{bk} S_{r+1}\right)} \quad (12)$$

Where $b_{r+1,d}$ represents emission probability of visible symbol bd that is emitted by hidden state Sr+1. This process continues until model is converged i-e variation of probability values in two consecutive iterations is within a specified threshold which is 0.09 in proposed technique. Training process of HMM is shown in fig. 13.

Classification Process

After training each alphabet character is associated to a separate HMM. Each test image is represented by two observation sequences containing feature coefficients that are slightly different with each other as given below.

$$b_1^T = \left[\sigma H, \sigma V, \sigma LD, \sigma RD, \sigma NH\right]$$

$$A = \begin{array}{c} \\ W_1 \\ W_2 \\ W_3 \\ W_4 \\ W_5 \\ W_6 \\ W_7 \\ W_8 \\ W_9 \end{array} \begin{array}{c} \begin{array}{ccccccccc} W_1 & W_2 & W_3 & W_4 & W_5 & W_6 & W_7 & W_8 & W_9 \end{array} \\ \begin{bmatrix} 0.6538 & 0.038 & 0.038 & 0 & 0.115 & 0.0384 & 0.115 & 0 & 0 \\ 0.667 & 0 & 0 & 0 & 0.333 & 0 & 0 & 0 & 0 \\ 0.5 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 \\ 0.1667 & 0.1667 & 0.1667 & 0.5 & 0 & 0 & 0 & 0 & 0 \\ 0.25 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0.125 & 0.125 \\ 0 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0.5 & 0 \\ 0.25 & 0 & 0 & 0.125 & 0 & 0 & 0.5 & 0.125 & 0 \\ 0.4 & 0 & 0 & 0.2 & 0 & 0 & 0.2 & 0.2 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

Fig. 12. Transition Probability Matrix

$b_2^T = [\sigma H * 1.01, \sigma V * 1.02, \sigma LD * 1.03, \sigma RD * 1.04, \sigma NH * 1.05]$

Most likelihood sequence of hidden states/character zones through which model has made transitions while generating observation sequence b_u^T is determined.

$$P\left(S^T \mid b_u^T\right) = ? \quad (13)$$

For this probability of each observation sequence of a test image is to be determined against each trained HMM using (14).

$$P\left(b_u^T \mid \theta_n\right) = ?, \; u = 1, 2 \quad (14)$$

Where θ_n represents the trained HMMs. To determine this probability, we find out all possible sequences of hidden states/character zones that may generate the given observation sequence. $P\left(b^T \mid \theta\right)$ is calculated by taking summation of probabilities of all these hidden state sequence using (15)

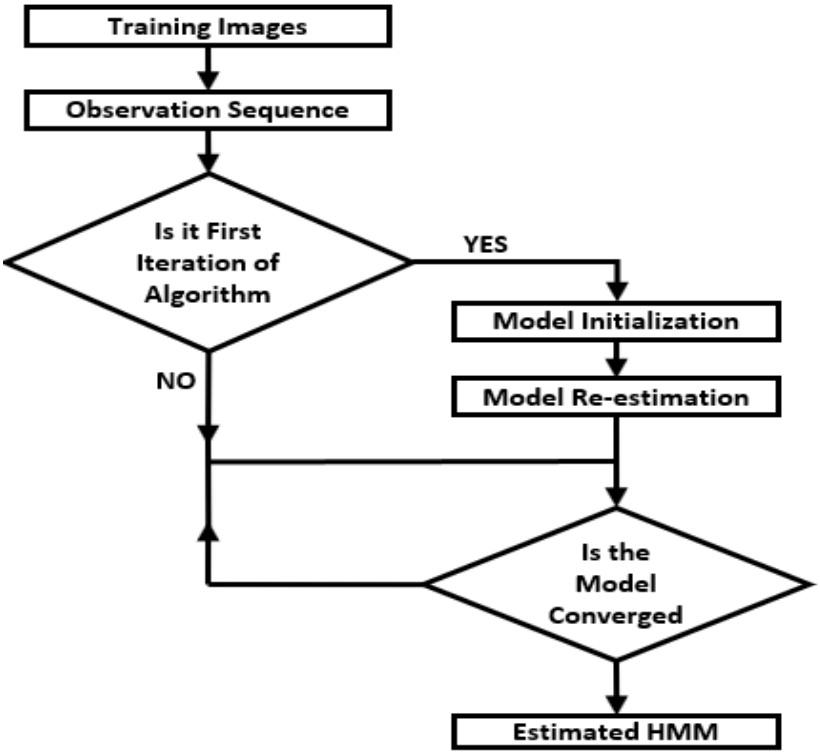


Fig. 13. Training process of HMM.

$$P\left(b_u^T \mid S_i^T\right).P\left(S_i\right), \qquad i\max = N_s^T \quad (15)$$

Where imax represents the maximum number of possible paths of hidden states through which model can make transitions while generating b_u^T .Hidden sequence S_i^T represents one of those possible hidden state sequences S_i^T of length T that has generated b_u^T .

$$S_1^T = \{S(1), S(2), S(3), \ldots\ldots\ldots, S(T)\} \quad (16)$$

Probability of a particular hidden sequence is given in (17) that is the product of transition probabilities at different time instances. Similarly, probability of observation sequence for a known hidden sequence is given in (18) that is the product of emission probabilities at different time steps.

$$P(S_1^T) = \pi_{t=1}^T P(S(t) \mid S(t-1)) \quad (17)$$

$$P(b_u^T \mid S_1^T) = \pi_{t-1}^T P(b_u(t) \mid S(t)) \quad (18)$$

For time instant t=1, $P(b_u^T \mid S_1^T)$ is the probability that model is at first state of a

particular hidden sequence and it has generated first visible symbol of known observation sequence of test image. By putting the values of (17) and (18) in (15) we get

$$P(b_u^T \mid \theta) = \sum_{i=1}^{i\max} \pi_{t=1}^{T} P(b_u(t) \mid S_i(t)). \\ P(S_i(t) \mid S_i(t-1)) \quad (19)$$

Once probability of an observation sequence of a test image against a trained HMM is obtained, most probable sequence of hidden states through which the model has made transitions while generating b_u^T is determined. So, at each time step the most probable hidden state is determined and this process continues until t= T where T is the length of observation sequence. The same process is repeated for second observation sequence of a test image. Test image is classified based on the majority vote rule of most likelihood sequence of hidden states for known observation sequences as given in (20).

$$I_{test} = \begin{cases} I_k & if\ P(S_k^T \mid b_u^T) \max_n P(S_n^T \mid b_u^T) \\ unknown & otherwise \end{cases} \quad (20)$$

Where $P(S_k^T \mid b_u^T)$ is the probability of hidden state sequence given observation sequence b_u^T .Test image I_{test} is classified to the kth character in the database if probability of hidden state sequence is maximum for both observation sequences.

Complexity Analysis

Complexity of HMM training process is proportional to complexity of Baulm-Welch algorithm. Baulm-Welch algorithm has time complexity per iteration of O (N_s^2T) [12]. Where N_s is the number of states of model and T is observation sequence length. Observation sequence length T in the proposed technique is calculated using (21)

$$T = N_I * N_{D/I} * N_{UD/I} * N_{B/I} * N_{F/B} \quad (21)$$

Where N_I is the number of images used in training, $N_{D/I}$ is the number of dimensions using DWT per training image $N_{UD/I}$ is the number of dimensions using Universe of Discourse per training image, $N_{B/I}$ is number of blocks per training image and $N_{F/B}$ is number of feature coefficients per block. Overall complexity of HMM training for all alphabet characters in the alphabet character database is calculated using (22).

$$overall\ Complexity = N_c * N_{it} * N_s^2 * T \quad (22)$$

Where N_c is the total number of characters and N_{it} is the number of iterations of Baulm-welch algorithm. Viterbi algorithm is used for testing or evaluation of an observation sequence. Complexity of Viterbi algorithm is O (Ns^2T) [17]. Overall complexity of HMM decoding for all characters in the database is $N_c * N_s^2 * T$.

Where N_c is the total number of characters and N_{it} is the number of iterations of Baulm-welch algorithm. Viterbi algorithm is used for testing or evaluation of an observation sequence. Complexity of Viterbi algorithm is O (Ns^2T) [17]. Overall complexity of HMM decoding for all characters in the database is $N_c * N_s^2 * T$.

Experimental Results

The proposed technique is implemented in C++ with MATLAB 2016b software on a system with an Intel(R) Core (TM) i3-4010U CPU @ 1.70 GHz 1.70 GHz, 4.00 GB RAM, 64-bit. Uppercase, lowercase English alphabet and numeral character databases are used to verify the effectiveness of the proposed algorithm in character recognition. Uppercase and lowercase English alphabet character database contains 520 hand drawn character images (A-Z and a-z) of 52 alphabets with different font types and sizes in bitmap format [31]. Numerals Character database contains 100 hand drawn numeral character images (0-9) of 10 numerals with different font types and sizes in bitmap format [31]. In fig. 14 example images of these databases are shown. Crossed images represent misclassification.

For each of these databases, two data sets have been generated. One data set is named as trained data set and the other is test data set.

Trained data set consists of the five images from every class and thus we have total 130 training images in case of uppercase English alphabet characters, 130 training images in case of lowercase characters and 50 training images in case of numeral characters. The Testing data set consists of the five images from every class and thus we have total 130 testing images in case of uppercase English alphabet characters, 130 testing images in case of lowercase English alphabet characters and 50 test images in case of numeral characters. Recognition accuracy of proposed model is calculated on these databases one by one using 5, 6, 7, 8 and 9 images of each subject for training the model and remaining images for testing that is depicted in table 1. The recognition rate increases or in other words, the accuracy of the model will be increased if the no. of trained images is increased and vice versa.

Fig. 14. Example images of uppercase, lowercase and numeral character databases

Tab. 1. Recognition accuracy of proposed model on uppercase, lowercase and numeral character database

Number of training images	Recognition rate (%) (Uppercase English alphabet characters database)	Recognition rate (%) (Lowercase English alphabet characters database)	Recognition rate (%) (Numeral characters database)
N_{tr}=5	95.38	94.61	96
N_{tr}=6	96.15	95.19	97.5
N_{tr}=7	98.7	96.15	100
N_{tr}=8	100	96.15	100
N_{tr}=9	100	98.07	100

The complexity analysis of the proposed algorithm has been done on 310-character images of uppercase, lowercase and numeral character databases. Table 2 and 3 depicts the complexity analysis of HMM training with and without universe of discourse, discrete wavelet transforms and singular value decomposition. Table 4 and 5 depicts the complexity analysis of HMM evaluation using Viterbi algorithm with and without UD, DWT and SVD. This analysis shows that considerable computational complexity reduction is achieved up to 0.8%.

Comparative Study

Table 6 summarizes the recognition accuracy of state of the art hand written character recognition techniques and proposed model on handwritten big English alphabet character datasets.

Conclusions

The evaluations and comparisons are performed on 260-character images of the uppercase English alphabet character database, 260-character images of the lowercase English alphabet character database and 100-character images of numerals character database. Very

promising results are achieved when geometric-based local features and the Hidden Markov Model along with Discrete Wavelet Transform is used to recognize the handwritten English alphabetical characters. The results showed an achievement of 96.15%, recognition rate when 50 % of images in total are used for training the HMM. This recognition accuracy increases as the source data set is enlarged. The complexity analysis of the proposed algorithm has been done using DWT, Universe of Discourse and SVD techniques and considerable computational complexity reduction has been achieved up to 0.8%.

Acknowledgments

N_I	$N_{D/I}$	$N_{UD/I}$	$N_{B/I}$	$N_{F/B}$	T	N_{IT}	Overall Complexity
310	50x50	50x50	9	9	65,812,500,000	5	693,005,625,000,000

Tab. 2. Complexity of HMM training without UD, DWT and SVD.

N_I	$N_{D/I}$	$N_{UD/I}$	$N_{B/I}$	$N_{F/B}$	T	N_{IT}	Overall Complexity	Computational Complexity in percentage
310	25x25	9x16	9	5	526,500,000	5	693,005,625,000,000	0.8%

Tab. 3. Complexity of HMM training with UD, DWT and SVD.

N_C	N_s^2	N_I	$N_{D/I}$	$N_{Ud/I}$	$N_{B/I}$	$N_{F/B}$	T	Overall Complexity
62	81	310	50x50	50x50	9	9	65,812,500,000	138,601,125,000,000

Tab. 4. Complexity of HMM Evaluation using Viterbi algorithm without UD, DWT and SVD.

N_C	N_s^2	N_I	$N_{D/I}$	$N_{Ud/I}$	$N_{B/I}$	$N_{F/B}$	T	Overall Complexity	Computational Complexity in percentage
62	81	310	25x25	9 x16	9	5	526,500,000	1,108,809,000,000	0.8%

Tab. 5. Complexity of HMM Evaluation using Viterbi algorithm with UD, DWT and SVD.

Ref	Publication year	Classifier used	features	Accuracy
[14]	2011	Decision tree	Horizontal and vertical Line Count and position	91%
[15]	2011	NN & SVM	Fourier Descriptors	62.93%
[16]	2012	ANN	Discrete Wavelet Transform (DWT)	98.46%
[17]	1995	HMM	Location, curve of edge & percentage of pixels lying on the edge	94.15%
[18]	2003	BP & RBF Networks	Directional and Transitional features	85.48%
[19]	2010	Feed forward Back propagation neural network	Diagonal feature	98%
[20]	2013	SVM	Image centroid zone, zone centroid zone, Hybrid centroid zone	95.11%

[21]	2013	Directional pattern matching	12 directional features	96.2%
[22]	2001	HMM&NN	140 geometrical features of every pre segmented frames	96.1%
Proposed				96.15%

This research has been supported by Image Processing cell at Military College of Signals, National University of Sciences and Technology, Islamabad.

About the Authors ...

NIMRA ASIF was born in 1995 (Pakistan). She is currently doing her MS from Military college of signals, NUST, Pakistan. She received her bachelor degree from university of engineering & technology Taxila. Her research interests include antenna desiginig and wave propagation.

Imran TAUQIR was born in 1965 (Pakistan). He received his PhD. in Electrical Engineering (Image Processing) from University of Engineering and Technology, Lahore, Pakistan. His research interests include Wavelet based image processing, sparse signal/ image coding, signal and image processing and stochastic process.

Adil-Masood SIDDIQUI completed his bachelor in telecommunications engineering in 199. Afterwards he pursue his Masters and PHD degree from University of Engineering and technology in 2005 and 2009 respectively. His research interest includes, image de-noising, image enhancement and defogging. He has done many research

REFERENCES

DISHANT, N., Feature Extraction Technique for Handwritten Character Recognition using geometric-based Artificial Neural Network. International Journal of Advance Engineering and Research Development, 2016, vol. 3, no. 8, p. 2348-6406, DOI: https://doi.org/10.21090/ijaerd.030818

AYYAZ, M., JAVED, et al. Handwritten character recognition using multiclass SVM classification with hybrid feature extraction. Pakistan Journal of Engineering and Applied Sciences, 2016, Vol. 10, p. 57-67

PATEL, D., SOM, et al. Improving the recognition of handwritten characters using neural network through multiresolution technique and euclidean distance metric. International Journal of Computer Applications, 2012, vol. 45, no. 6, p. 38–50

PATEL, C., RIPAL, et al. Handwritten character recognition using neural network. International Journal of Scientific \& Engineering Research, 2011, vol. 2, no. 5, p. 1–6

TAPPERT, C., SUEN, et al. The state of the art in online handwriting recognition. IEEE Transactions on pattern analysis and machine intelligence, 1990, vol. 12, no. 8, p. 787–808 DOI: 10.1109/34.57669

BLUMENSTEIN, M., VERMA, et al, A novel feature extraction technique for the recognition of segmented handwritten characters. Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings, 2003, p. 137–141 DOI: 10.1109/ICDAR.2003.1227647

BECERIKLI, Y., OLGAC, et al. Neural network-based license plate recognition system. 2007 International Joint Conference on Neural Networks, 2007, p. 3057–3061, DOI: 10.1109/IJCNN.2007.4371448

HAMAMOTO, Y., UCHIMURA, et al. Recognition of handprinted Chinese characters using Gabor features. Proceedings of 3rd International Conference on Document Analysis and Recognition, 1995, vol. 2, p. 819–823, DOI: 10.1109/ICDAR.1995.602027

KIM, K., PARK, et al. Recognition of handwritten numerals using a combined classifier with hybrid features. Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR), 2004, p. 992–1000, DOI: https://doi.org/10.1007/978-3-540-27868-9_109

ALI, S., GHANI, et al. Handwritten digit recognition using DCT and HMMs. 2014 12th International Conference on Frontiers of Information Technology, 2014, p. 303–306, DOI: 10.1109/FIT.2014.63

RASHID, S., Optical Character Recognition-A Combined ANN/HMM Approach. 2014

ABUZNEID, M., MAHMOOD, et al. Enhanced human face recognition using LBPH descriptor, multi-KNN, and back-propagation neural network. IEEE access, 2018, vol. 6, p. 20641–20651, DOI: 10.1109/ACCESS.2018.2825310

JENA, O., PRADHAN, et al, Hidden Markov Model Based Odia Numeral Recognition Using Moment and Structural. International Journal of Engineering and Advanced Technology, 2019, vol. 8, no. 6, ISSN: 2249 – 8958, DOI: 10.35940/ijeat. F8614.088619

RAHIMAN, M., RAJASREE, et al. Recognition of handwritten Malayalam characters using vertical \& horizontal line positional analyzer algorithm. 2011 3rd International Conference on Electronics Computer Technology, 2011, vol. 2, p. 268–274

GUPTA, A., SRIVASTAVA, et al. Offline handwritten character recognition using neural network. 2011 IEEE International conference on computer Applications and industrial Electronics (ICCAIE), 2011, p. 102–141

Patel, D., T., et al. Improving the Recognition of Handwritten Characters using Neural Networkthrough Multiresolution Technique and Euclidean Distance Metric. International Journal of Computer Applications, 2012, vol. 45, no. 6, p. 38-50

BUNKE, H., ROTH, et al. Off-line cursive handwriting recognition using hidden markov models. Pattern recognition, 1995, vol. 28, no. 9, p. 1399–1413

BLUMENSTEIN, M., VERMA, et al. A novel feature extraction technique for the recognition of segmented handwritten characters. Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings, 2003, p. 137–141

PRADEEP, J., SRINIVASAN, et al. Diagonal feature extraction based handwritten character system using neural network. International Journal of Computer Applications, 2010, vol. 8, no. 9, p. 17–22

SINHA, G. Arabic numeral Recognition Using SVM Classifier. International Journal of Emerging Research in Management \& Technology ISSN, 2013, p. 2278–9359

HALLALE, S., SALUNKE, et al. Twelve directional feature extraction for handwritten English character recognition. International Journal of Recent Technology and Engineering, 2013, vol. 2, no. 2, p. 39–42

TAY, Y., LALLICAN, et al. An offline cursive handwritten word recognition system. Proceedings of IEEE Region 10 International Conference on Electrical and Electronic Technology. TENCON 2001 (Cat. No. 01CH37239),2001, vol. 2, p. 519–524

MIARNAEIMI, H., DAVARI, et al. A New Face Recognition System Using HMMs Along with SVD Coefficients. Iranian Journal of Electrical and Electronic Engineering, 2008, DOI: 10.5220/0001072002000205

DEEPA, A., RAO, et al. Feature extraction techniques for recognition of Malayalam handwritten characters. International Journal Advanced Trends Computer Science Engineering, 2014, vol. 3, p. 481–485

SIDDHARTH, K., JANGID, et al. Handwritten Gurmukhi character recognition using statistical and background directional distribution. International Journal Advanced Trends Computer Science Engineering, 2011, vol. 3, no. 6, p. 2332–2345

KUMAR, S. Neighbourhood pixels weights-a new feature extractor. International Journal of Computer Theory and Engineering, 2010, vol. 2, no. 1, p. 69, DOI: 10.7763/IJCTE. 2010.V2.119

KIM, K., PARK, et al. Recognition of handwritten numerals using a combined classifier with hybrid features. Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition, 2004, p. 992–1000, DOI: https://doi.org/10.1007/978-3-540-27868-9_109

ARICA, N., YARMAN-VURAL, et al. Optical character recognition for cursive handwriting. IEEE transactions on pattern analysis and machine intelligence, 2002, vol. 24, no. 6, p. 801–813, DOI: 10.1109/TPAMI.2002.1008386

PAPAKOSTAS, G., KOULOURIOTIS, et al. Feature extraction based on wavelet moments and moment invariants in machine vision systems. Human-Centric Machine Vision, 2012, p. 31, DOI: https://doi.org/10.5772/33141

SINGH, A., NATH, et al. Handwritten English character recognition using HMM Baum-welch and genetic algorithm. International Journal of Computer Science and Information Technologies, 2016, vol. 7, no. 4, p. 1788–1794 http://www.mathworks.com/matlabcentral/fileexchange/26158-hand-drawn-sample-images-of-english-characters

[32] GHADIYARAM, A., An investigation into Telugu font and character recognition. Hyderabad

LEE, S., KIM, et al. Multiresolution recognition of handwritten numerals with wavelet transform and multilayer cluster neural network. Proceedings of 3rd International Conference on Document Analysis and Recognition, 1995, vol. 2, p. 1010–1013 DOI: 10.1109/ICDAR.1995.602073

ALOBAIDI, T., MIKHAEL, et al. Face recognition system based on features extracted from two domains. 2017 IEEE 60th International Midwest Symposium on Circuits and Systems, 2017, p. 977–980, DOI: 10.1109/MWSCAS.2017.8053089

Nath, R., KUMAR, et al. Handwritten English character recognition using HMM, Baum-welch and genetic algorithm. 2016

SANDHYA, R., KRISHNAN, HMM based – character recognition for offline handwritten Indic scripts. International journal of engineering research and technology, 2015, vol. 4, DOI: https://doi.org/10.17577/ijertv4is060810

KIM, H., DAIJIN, et al. A numeral character recognition using the PCA mixture model. Pattern Recognition Letters, 2002, vol. 23, no. 1-3, p. 103–111 DOI: 10.1016/s0167-8655(01)00093-9

VITHLANI, P., KUMBHARANA, et al. A Study of Optical Character Patterns identified by the different OCR Algorithms. International Journal of Scientific and Research Publications, 2015, vol. 5, no. 3, p. 2250–3153

KIM, H., DAIJIN, et al. A numeral character recognition using the PCA mixture model. Pattern Recognition Letters, 2002, vol. 23, no. 1-3, p. 103–111 DOI: 10.1016/s0167-8655(01)00093-9