American Journal of Electrical and Electronic Engineering, 2015, Vol. 3, No. 2, 51-63
Available online at http://pubs.sciepub.com/ajeee/3/2/5
© Science and Education Publishing
DOI:10.12691/ajeee-3-2-5

Design Proposed Features Extraction Recognition System of Latin Handwritten Text Based on 3D-Discrete Multiwavelet Transform

Laith Ali Abdul-Rahaim*

Electrical Engineering Department, Babylon University, Babil, Iraq
*Corresponding author: drlaithanzy@yahoo.com

Received April 02, 2015; Revised April 15, 2015; Accepted April 20, 2015

Abstract On-line handwriting recognition is the task of determining what letters or words are present in handwritten text. It is of significant benefit to man-machine communication and can assist in the automatic processing of handwritten documents. It is a subtask of the Optical Character Recognition (OCR), whose domain can be machine-print only.The introduced system is a character-based recognition and it is a writer independent system. The recognition responsibility of the proposed system is for 52 character classes [uppercases (A-Z) and the lowercases (a-z)]. The suggested system includes the essential stages needed for most of the pattern recognition systems. These stages are the preprocessing stage, the features extraction stage, the pattern matching and classification stage and the postprocessing stage. The proposed method employs the 3 Dimensional Multiwavelet transform 3D-DMWTCS using multiresolution image decomposition techniques working together with multiple classification methods as a powerful classifier. The classification stage is designed by using a minimum distance classifier depending on Euclidean Distance which has a high speed performance. The system design also includes a modest postprocessing stage that makes a consistency between the recognized characters within the same word in relation to their upper and lower cases.The overall classification accuracy of proposed systems can be obtained are 95.305 percent with 3D-DMWTCS based on the Rimes database.

Keywords: *3D-DMWTCS,2D-DMWTCS, dpi, HWR, RR, OCR, ED, MDC*

Cite This Article: Laith Ali Abdul-Rahaim, "Design Proposed Features Extraction Recognition System of Latin Handwritten Text Based on 3D-Discrete Multiwavelet Transform." *American Journal of Electrical and Electronic Engineering*, vol. 3, no. 2 (2015): 51-63. doi: 10.12691/ajeee-3-2-5.

1. Basic Concepts of the Handwriting Recognition

Handwriting (HW) is one of the most important ways in which civilized people communicate. It is used for both personal (e.g. letters, notes, addresses on envelopes, etc.) and business communications (e.g. bank cheques, business forms, etc.). The writing is a physical process where the brain sends an order through the nervous system to the arm, hand and fingers, where together they manipulate the writing tool. Therefore, a person's handwriting is as unique as human fingerprints and facial features. However, it varies depending upon many factors (age, education, temper, left or right handed writer, etc.).With the advent of the computer, it became possible that machines could also reduce the amount of mental labor needed for many tasks. One of these tasks is recognizing a human's handwriting. Of course, much progress has been made in the way of computer handwriting recognition, but a computer will never be able to read a human's handwriting as good as a human. Even so, it doesn't hurt to try to develop technology which can approach the recognition ability of humans. Since the handwriting is one of the most important ways in which people communicate, it would provide an easy way of interacting with a computer [1,2]. A recognition system can be either "on-line" or "off-line." It is "online" if the temporal sequence of points traced out by the pen is available, such as with electronic personal data assistants that require the user to "write" on a screen where the path of the pen is measured by a device such as a digitizing tablet. It is "off-line" if it is applied to previously written text, such as any image scanned by a scanner. The on-line problem is usually easier than the off-line problem since more information is available. So far, most of the off-line handwriting recognition systems are applied to reading letters, postal addresses and then automatic sorting of postal mail, processing forms like bank cheques or discrimination of the different scripts for individual writers (Handwriting identification) [3]. The real progress in character recognition was achieved in the advancement age (after 1990). In early 1990s, image processing and pattern recognition were efficiently combined with artificial intelligence techniques. Efficient tools such as Neural Networks (NN), Support Vector Machines (SVM), Hidden Markov Model (HMM), fuzzy set reasoning, and natural language combined with more powerful computers and more accurate electronic equipment's have provided quite satisfactory results for restricted applications [4]. The challenge is to recognize the HW texts that are written by people in real life situations and makes an automatic transcription by computer, where only the image of the handwriting is available. It becomes more important to make the transfer of information between people and machines simple, fast and reliable. Chevalier et al [5] presented a two-dimensional approach of the processing of handwriting. It combines a Markovian model, an efficient decoding algorithm, a windowed spectral features extraction scheme and a rigorous evaluation methodology. They applied this principle to a digit recognition task and to a word recognition task [6].

2. Model for on-line Handwriting Recognition

Handwriting Recognition is interpretation of data which describes handwritten objects to generate a description of that interpretation in a desired format. Or in other words it is a determination what letters or words are present in a digital image of handwritten text. HWR is of significant benefit to man-machine communication and can assist in the automatic processing of the handwritten documents. A wide variety of techniques are used to perform off-line handwriting recognition. To convert this image into information understandable by computers requires the solution to a number of challenging problems. Firstly, pre-processing steps are achieved on the image to reduce some undesirable variability that only contributes to complicate the recognition process. Operations like binarization, noise removal, skew, slant and slope corrections, thinning, smoothing, normalization, etc. are carried out at this stage. The second step is the segmentation of the word into a sequence of basic recognition units such as characters or pseudo-characters. However, segmentation may not be present in all systems. Recognition approaches can be either “Holistic” or segmentation-based. “Holistic” means that words or sentence are processed as a whole without segmentation into characters or strokes [6,7]. In segmentation-based approaches, whole or partial characters are recognized individually after they have been extracted from the text image. The final step is to extract discriminated features from the input pattern to either build up a feature vector or to generate graphs, string of codes or sequence of symbols. However, the characteristics of the features depend on the preceding steps; say whether segmentation of words into characters was carried out or not. The pattern recognition model to handwriting recognition consists of pattern training, that is, one or more patterns corresponding to handwritten words or characters of the same known class are used to create a pattern representative of the features of that class. The recognition includes a comparison of the test pattern with each class reference pattern and measuring a similarity score (e.g. distance, probability) between the test pattern and each reference pattern [8,9]. The pattern similarity scores are used to decide which reference pattern best matches the unknown pattern. The post-processing or verification may also be included in some systems. However, for meaningful improvements in recognition, it is necessary to incorporate the recognition process other sources of knowledge such as language models [10]. A limited vocabulary is one of the most important aspects of systems that rely on large vocabularies because it contributes to improve the accuracy as well as to reduce computation. In the case of systems that deal with large vocabularies, other additional modules may be included such as pruning or lexicon reduction mechanisms [11,12]. If we desire a system to distinguish objects of different types, it must be first decided which characteristics of the objects should be measured to produce descriptive parameters called (features) of the object, and the resulting parameters values comprise the feature vector for each object. Proper selection of the features is important, since only these will be used to identify the objects.Good features have four characteristics [13,14]; Discrimination, Reliability, Independence and Small Numbers. The complexity of a pattern recognition system increases rapidly with dimensionality of the system. More importantly, the number of objects required to train the classifier and to measure its performance increases exponentially with the number of features. And it is necessary to avoid the redundancy.The features extraction is an important step in achieving good performance for the recognition. For off-line HWR systems the feature extraction methodology is based on one or more of Global features and Geometrical and topological features. The geometrical and topological (structural) features describe the geometry and topology characteristic of a character. Some examples of extracted features are strokes and bays (kerning) in various directions, dots, end points, intersections of line segments (junctions), loops, curves (turnings), etc as shown in Figure 1. Each of these features can be encoded by a single number. Geometrical and topological features have a high tolerance to distortion and style variations. Due to complexity of extracting the geometrical and topological features and the great variations in local properties of HW characters, it is rather difficult to generate feature masks. But once they are implemented, they can process characters at high speed independently in [15] are examples of using the geometrical features.

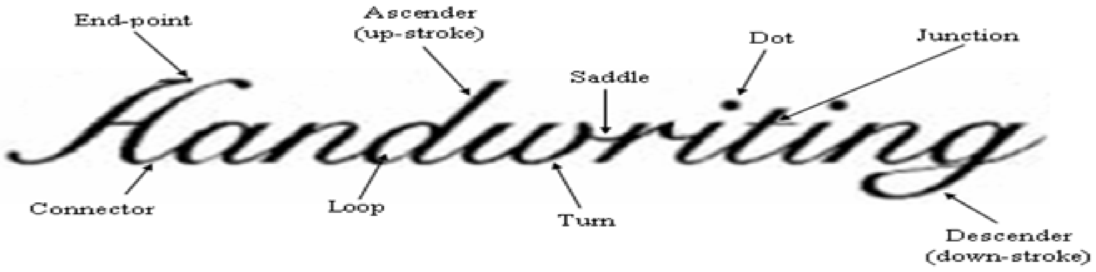


Figure 1. Geometrical and topological features

Statistical features: The statistical features are derived from the statistical distribution of pixels of a character. They are numerical measures computed over images or regions of images [16,17]. They include, but are not limited to, pixel densities, histograms of chain code directions, moments, Fourier descriptors, the aspect ratio of the character, characteristic loci, crossing and distances. The statistical features take some topological and dynamic information into account and consequently can tolerate minor distortions and style variations [18,19]. A key question in handwriting recognition is how test and reference patterns are compared to determine their similarity. Pattern comparison can be done in a wide variety of ways. The goal of a classifier is to match a sequence of observations derived from an unknown handwritten word against the reference patterns that were previously trained, and obtain confidence scores (distance, cost, or probabilities) to further decide which model best represents the test pattern [20]. Contextual approaches lead to excessive growth in the number of models since one model is needed for each pair of adjacent characters. Parameter estimation may be unreliable since, for practical applications, a restricted set of training data is generally available. It is thus desirable to reduce the number of models and model parameters while preserving model refinement. Hence, model sharing and parameter tying are necessary to reduce the number of parameters. Schussler and Niemann [21] describe a context-dependent system using HMMs, where all sub word units (from monographs to the whole word) are modeled within a word hierarchy. Models with not enough training samples are eliminated. The system is tested on a small and dynamical lexicon. The state-based tying proposed by Natarajan et al.[22] uses a mixture of 128 Gaussians associated to each state position of contextual models (trigraphs) corresponding to the same base character. The total number of models can also be reduced as in [23,24] by clustering all trigraphs according to contexts described not as characters but as ascending or descending strokes. Fink and Plotz [25] also proposed a system based on contexts, these contexts being defined as broad categories. A data-driven clustering of the 1,500 Gaussian densities of the mixture is performed at each state position and for each category. It is worth noting that clustering and tying the contextual models may offer, in addition to reducing the number of parameters, the possibility to automatically capture common contextual effects on handwriting a given letter [26].

3. Feature Extraction Using Multiwavelet Transform and Classification

Multiwavelet transform is a new concept in the framework of wavelet transform but has some important differences. In particular, whereas wavelet has an associated one scaling function and wavelet function, Multiwavelet has two or more scaling functions and wavelet functions. One of the well-known Multiwavelet was constructed by Donovan, Geronimo, Hardin, and Massopust (DGHM). DGHM Multiwavelet simultaneously possesses orthogonality, compact support, approximation order 2 and symmetry [13]. Next, we give a brief overview of the Multiwavelet transform [14]. Unlike scalar wavelet, even though the Multiwavelet is designed to have approximation order *p,* the filter bank associated with the Multiwavelet basis does not inherit this property. Furthermore, since the Multiwavelet have more than one scaling function; the dilation equation becomes dilation with matrix coefficients. Thus, in applications, one must associate a given discrete signal into a sequence of length *r* vectors (where *r* is the number of scaling functions) without losing certain properties of the underlying Multiwavelet. Similar to the traditional scalar wavelet transform, the two-dimensional Multiwavelet transform can be achieved by applying the one-dimensional transform on the rows by treating each row as a one-dimensional signal and afterward on columns. However, for the applications using Multiwavelet, profiteering process must be applied to each row and each column to initiate the vector sequence *c0* to the filter bank.. There is growing interest in using wavelet features for images in many applications, including object identification, medical-images retrieval, and texture analysis. These features are generally extracted from the two-dimensional discrete Multiwavelet (2D-DMWTCS) coefficients of the image under processing. Wavelets are localized basis functions which are translated and dilated versions of some fixed mother wavelet. The main feature of wavelets is that they are able to provide localized frequency information about a function or signal. Such information is particularly beneficial for classification. There exists an abundant variety of wavelets, and the fundamental problem to overcome is deciding which wavelet will produce the best results for a particular application. The following procedure must be doing to calculate a single level 2-D Discrete Multiwavelet Transform using GHM four multifilter and a critical sampled scheme of preprocessing (approximation row preprocessing):

1. *Checking phoneme Signal Dimensions:* Phoneme matrix should be a square matrix, N×N matrix, where N must be power of 2. So that the first step of the transform procedure is checking input phoneme dimensions. If the phoneme matrix is not a square matrix, some operation must be done to the adding rows or column of zeros to get a square matrix.
2. *Constructing A Transformation Matrix:* Using the transformation matrix, such as given in the following matrix format [10]:

$$\begin{aligned}&\text{new odd-row}\\&=(0.373615)[\text{same odd-row}]\\&+(0.11086198)[\text{next even-row}]\\&+(0.11086198)[\text{previous even-row}]\end{aligned} \tag{1}$$

$$\text{new even-row}=(\sqrt{2}-1)[\text{same even-row}] \tag{2}$$

An N/2×N/2 transformation matrix should be constructed using GHM low-and high-pass filtersmatrices given below [9]:

$$\begin{bmatrix} H_0 & H_1 & H_2 & H_3 & 0 & 0 & \dots \\ G_0 & G_1 & G_2 & G_3 & 0 & 0 & \dots \\ 0 & 0 & H_0 & H_1 & H_2 & H_3 & \dots \\ 0 & 0 & G_0 & G_1 & G_2 & G_3 & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \end{bmatrix} \tag{3}$$

$$\left.\begin{aligned}
H_0 &= \begin{bmatrix} \frac{3}{5\sqrt{2}} & \frac{2}{5} \\ -\frac{1}{20} & -\frac{3}{10\sqrt{2}} \end{bmatrix}, H_1 = \begin{bmatrix} \frac{3}{2\sqrt{2}} & 0 \\ \frac{9}{20} & \frac{1}{\sqrt{2}} \end{bmatrix} \\
H_2 &= \begin{bmatrix} 0 & 0 \\ \frac{9}{20} & -\frac{3}{10\sqrt{2}} \end{bmatrix}, H_3 = \begin{bmatrix} 0 & 0 \\ -\frac{1}{20} & 0 \end{bmatrix} \\
G_0 &= \begin{bmatrix} -\frac{1}{20} & -\frac{3}{10\sqrt{2}} \\ \frac{1}{10\sqrt{2}} & \frac{3}{10} \end{bmatrix}, G_1 = \begin{bmatrix} \frac{9}{20} & -\frac{1}{\sqrt{2}} \\ -\frac{9}{10\sqrt{2}} & 0 \end{bmatrix} \\
G_2 &= \begin{bmatrix} \frac{9}{20} & -\frac{3}{10\sqrt{2}} \\ \frac{9}{10\sqrt{2}} & -\frac{3}{10} \end{bmatrix}, G_3 = \begin{bmatrix} -\frac{1}{20} & 0 \\ -\frac{1}{10\sqrt{2}} & 0 \end{bmatrix}
\end{aligned}\right\} \quad (4)$$

3. After substituting GHM matrix filter coefficients values as given in eq. (3), an N×N transformation matrix results with the same dimensions of input image dimensions after preprocessing. Preprocessing Rows: Approximation-based row preprocessing can be compute by applying Eqs. (1), and (2) to the odd- and even-rows of the input N´N matrix respectively. Input matrix dimensions after row preprocessing are the same N´N.
4. *Transformation of image Rows :* The procedure can be done as follows:

a. Apply matrix multiplication between the N×N constructed transformation matrix by the N×N row preprocessed input phoneme matrix.

b. Permute the resulting N×N matrix rows by arranging the row pairs 1, 2 and 5, 6 …, N–3, N–2 after each other at the upper half of the resulting matrix rows, then the row pairs 3, 4 and 7, 8… N–1, N below them at the next lower half.

5. *Preprocess Columns:* To repeat the same procedure used in preprocessing rows:

a. Transpose the row transformed N×N matrix resulting from step 4.

b. Repeat step 3 to the N×N matrix (transpose of the row transformed N×N matrix), which results in N×N column preprocessed matrix.

6. *Transformation of Columns :* Transformation of phoneme columns is applied next to the N×N column preprocessed matrix as follows:

a. Apply matrix multiplication between the N×N constructed transformation matrixes by the N×N column preprocessed matrix.

b. Permute the resulting N×N matrix rows by arranging the row pairs 1, 2 and 5, 6 …, N–3, N–2 after each other at the upper half of the resulting matrix rows, then the row pairs 3, 4 and 7, 8… N–1, N below them at the next lower half.

7. *The Final Transformed Matrix:* The following procedure must be doing to get final transformation matrix:

a. Transpose the resulting matrix that get from column transformation step.

b. Apply coefficients permutation to the resulting transpose matrix. Coefficients permutation is apply to each of the basic four subbands of the resulting transpose matrix so that each subband permutes rows then permutes columns. Finally, a N×N DMWT matrix results from the N×N original matrix using approximation-based preprocessing.

The main features of this Multiwavelet type are the ability to provide localized frequency information about a character image. Such information is particularly beneficial for classification. For a given binary image containing a single character, there are many pre-processing procedures performed prior to feature extraction. The most important thing is to make our system independent of each character concerning its *shape*, *position (location in the word)* and *size*. In relation to its shape normalization, this can be achieved by slant and skew corrections steps. Concerning stroke width normalization, this is done by skeletonization (thinning) approach and successive steps of stroke thickening. These steps leave each stroke with approximately the same width. Related to character position normalization, it is achieved by first character segmentation and then a bounding rectangle of each character is found. This removes any differences due to location of character within each image. Next this bounding rectangle is scaled to a (32 × 32) pixel image (A^{j+1}), in order to scale (size) normalization. The wavelet decomposition is applied at one level of resolution, yielding four subband images {Approximations (A^j), Horizontal details (D_H^j), Vertical details (D_V^j) and Diagonal details (D_D^j)} each containing 16 x 16 pixels. Therefore, the feature vector is formed by these subband images with (*1xd*) dimensions, where *d*= 4 x 16 x 16. Figure 2 illustrates the 1-level of 3D-(DMWTCS or DWT) step. For each subband image, the values of the (wavelet or Multiwavelet) coefficients are normalized to the range [0, 1]. Figure 3 shows the 3D-(DMWTCS or DWT) coefficients for all subbands [19]. The main information is concentrated at the approximation subband and the other are distributed in the other subband images. Other experiments were done to find out which subband contains more important characteristics (important features for recognition).These experiments used the same test data set and the same data base. The results show the features relevance to the recognition process. The features that are extracted from the approximation subband contribute by about 53% of the importance in the recognition task, while the features extracted from the all other subbands contribute with about 47% of the significance in the recognition process. But this conclusion does not mean that the correct recognition is about 53 % by using the approximation only.

2.1. A Proposed Computation Method of 3D-DMWTCS

Discrete Multiwavelet transform given a good indication in applications of signal processing. Recent work on Multiwavelet have been studies the basic theory, methods of constructing new multifilters and the denoising and compression applications in of video and image [12,17,21,22]. The algorithms for computing three dimensional discrete Multiwavelet transform Critically

Sampled (3D-DMWTCS) have been described in this section in a simple and easy way to verify procedure using matrix multiplication and addition. To compute 3D-DMWTCS one must know how compute 1, 2D- discrete Multiwavelet transforms Critically Sampled and its inverse 1,2D-IDMWTCS. In 3D- discrete Multiwavelet transformation Critically Sampled algorithm is defined in 3D, so the transformation procedure will done successively in *x*-, *y*- and *z*-directions.

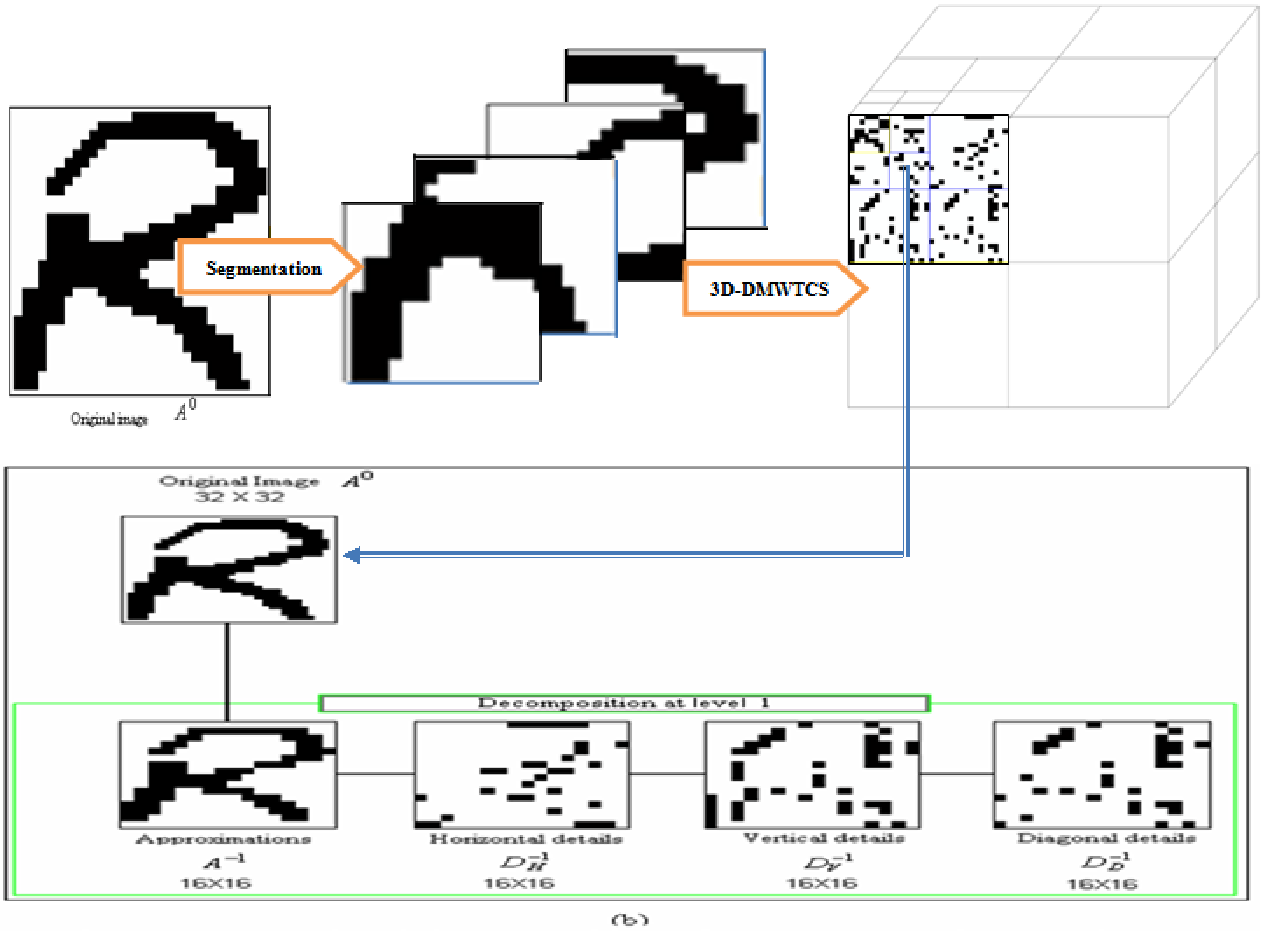


Figure 2. decomposition of 3D-DMWTCS on character imagea-square view mode b- Tree view mode

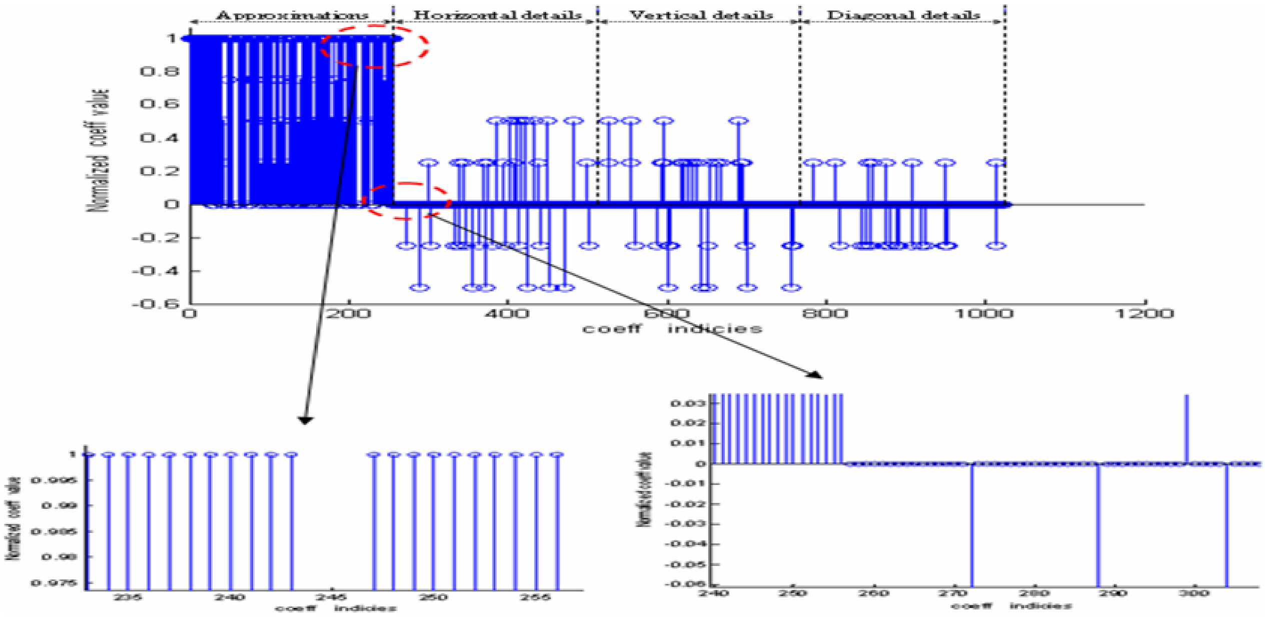
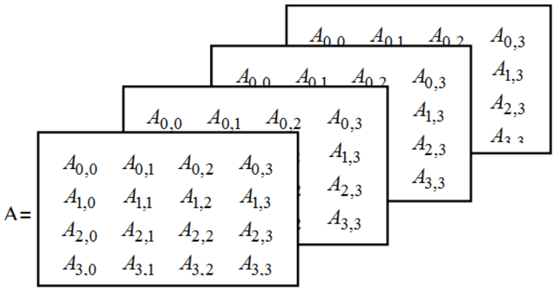


Figure 3. 3D-DMWTCS normalized coefficients

For a 2D-DMWTCS, the procedure was applied to each vector in *x*-direction first, and then to each vector in *y*-direction. Similarly, in 3D- discrete Multiwavelet transformation Critically Sampled the procedure is defined in 3D and the transformation algorithm is applied successively in *x*-, *y*- and *z*-direction.

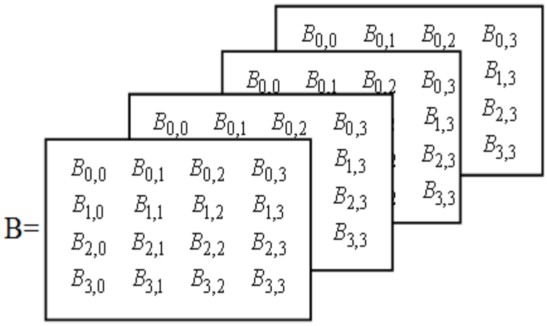
Let's take a general 3D signal, for example any NxNxM matrix. The computation 3D-DMWTCS need the following procedure:

1. construct 3D-matrix *A* to represent the 3D input signal,


A =
$A_{0,0}$ $A_{0,1}$ $A_{0,2}$ $A_{0,3}$
$A_{1,0}$ $A_{1,1}$ $A_{1,2}$ $A_{1,3}$
$A_{2,0}$ $A_{2,1}$ $A_{2,2}$ $A_{2,3}$
$A_{3,0}$ $A_{3,1}$ $A_{3,2}$ $A_{3,3}$

(5)

2. Using 2D DMWTCS algorithm to each NxN input matrix, which result in a Y matrix ($N\times N\times M$).


B=
$B_{0,0}$ $B_{0,1}$ $B_{0,2}$ $B_{0,3}$
$B_{1,0}$ $B_{1,1}$ $B_{1,2}$ $B_{1,3}$
$B_{2,0}$ $B_{2,1}$ $B_{2,2}$ $B_{2,3}$
$B_{3,0}$ $B_{3,1}$ $B_{3,2}$ $B_{3,3}$

(6)

3. Using 1D-DMWTCS algorithm shown in [11] tocompute 1D-DMWTCS to each element of *NxN*matrix in all *M* matrices in z-direction, which can summarized as follows:

a. For each i,j element in the 1st matrix construct a vector of *Mx1* for each element in z-direction output matrices from 2D-DMWTCS in step 2, this operation is done as below:

$$V(i,j) = \begin{bmatrix} B_{i,j,0} & B_{i,j,1} & B_{i,j,2} & B_{i,j,3} \end{bmatrix}^{T}_{1\times M} \qquad (7)$$

Where $i, j = 0,1,2,\cdots,N$.

b. Applying 1D-DMWTCS algorithm to each the construct vector $V(i,j)$.

4. Repeat step 3 for all construct vector i, j.
5. Finally, a$N\times N\times M$ of 3D-DMWTCS matrix results from the N×N×M original matrix using 3D-DMWTCS.

4. Training and Testing Phases of the Recognition Task

The goal of the training phase is to extract and prepare the best parameter values (features) of the character models. The training phase deals with the handwritten characters (some times are called letters) with known and defined letters classes. Each input letter image is adequately pre-processed and its relevant features are then extracted from the preprocessed image forming a feature vector (f_{dx1}), where *d* is the features number. For each character class, the feature vectors is generated which are also known as the class reference feature vectors. These vectors have the goal of representing its corresponding character classes. This class reference feature vectors are generated by the application of 2D-DMWTCS. As a result, the system has reference feature vectors which are forming the feature matrix ***F*** as stated in equation (8). ***F***contains*Mx52* feature vectors *f* where*M* is the number of training samples and (*52*) is the number of letter classes (upper lower cases of Latin alphabet). For example: $f_{1,1}$ and $f_{2,1}$are the feature vectors of the letter (A) of the first and second training samples and so on while $f_{1,2}$ and $f_{2,2}$ are the feature vectors of the letter (B) of the first and second training samples and so on. Each column in ***F*** represents the features of one class for *M* training samples. The size of the feature matrix depends on the feature vector dimension and the number of the available training samples *M*.

$$F = \begin{bmatrix} f_{1,1} & f_{1,2} & \cdots & \cdots & f_{1,52} \\ f_{2,1} & f_{2,2} & \cdots & \cdots & f_{2,52} \\ . & . & . & . & . \\ . & . & . & . & . \\ f_{M,1} & f_{M,2} & \cdots & \cdots & f_{M,52} \end{bmatrix} \qquad (8)$$

An extended training phase, i.e., more samples of handwritten letters with various styles would improve the system performance. The training data set have to be with various styles rather than to be of large quantity. The testing phase of the classification, an image from an unknown class is initially pre-processed and then having its feature vector extracted using the same techniques in the training phase. From this initially extracted feature vector, for each distinct letter class, a new and different feature vector is then generated. Next, the classifier is used to assign the input handwritten character to the class that best accommodates the input image.

4.1. The Minimum Distance Classifier (MDC)

The implemented minimum distance classification based on calculating and comparing the Euclidean Distances (*ED*). The Euclidean Distances are between the feature vector of the unknown input character (to be classified) and the reference feature vectors as shown in equation (9) [19].

$$ED_n = \left\| f_i - f_{m,n} \right\| = \sqrt{\sum_{m=1}^{M} \left| f_i - f_{m,n} \right|^2} \qquad n = 1,\ 2\ \ldots\ 52 \quad (9)$$

Where $\left\| f_i - f_{m,n} \right\|$ denotes the Euclidean Distance between the vectors f_i and $f_{m,n}$, f_i is the feature vector of the unknown input character pattern, the subscript *i* is denotes to the word "input". $f_{m,n}$is the m^{th} feature vector of the n^{th} character class that belongs to the feature matrix ***F***.

$$CED_n = \sum_{m=1}^{M} ED_{m,n} \qquad n = 1,\ 2\ldots\ 52 \qquad (10)$$

$$\mathrm{CED} = [CED_1\ CED_2\ \ldots\ldots CED_{52}] \qquad (11)$$

where CED_n is the Class Euclidean Distance of the n^{th} character class for *M* training samples and **CED** is the Class Euclidean Distance vector. The chosen class will be the one that achieves the smallest *CED* in **CED** In other words; it is the smallest distance between the input feature vector and the most representative (nearest) vector of the reference feature vectors. Figure 4 and Figure 5 show some examples of how minimum distance classifier work to classify the HW letters T&Z which appear at the left bottom corner of each plot. The x-axes of the plots represent the 52 characters classes of upper and lower cases. The y-axes are the *EDs* between the input features vector and the reference features vectors as shown in equation (11).

Concerning Figure 4, the CED_{20} is the smallest among all *CEDs*. The CED_{20} is the sum of *M* Euclidean Distances of the 20th class that belongs to the letter " T ", therefore, the input letter will be classified as T and so on for the other inputs. Figure 5 shows the letter "s-lowercase" classification. It is clear there are two smallest *CEDs*(CED_{19} *and* CED_{45}). The letter "s-lowercase" will be classified as "S-uppercase" since $CED_{19}<CED_{45}$. The recognition between letters with approximately having similar patterns of their upper and lower cases is still an open problem till this point of the proposed recognition system. Such letters are "s" and "S", "w" and "W", "z" and "Z" "c" and "C" etc. The unique difference between their upper and lower patterns is the size. The size difference is lost by the size normalization step at the preprocessing stage which may waste this feature between upper and lower cases of the letters above. This problem will be discussed and partially treated at the postprocessing stage.The false classification may come from calculating the smallest *ED* which may give an assurance to a wrong class as shown in Figure 6. Figure 7 shows the avoiding of the false classification for the HW letter "Y" as letter V. It is clear that there is some similarity between the HW "Y" with the letter V which is the reason for this very small *ED*.

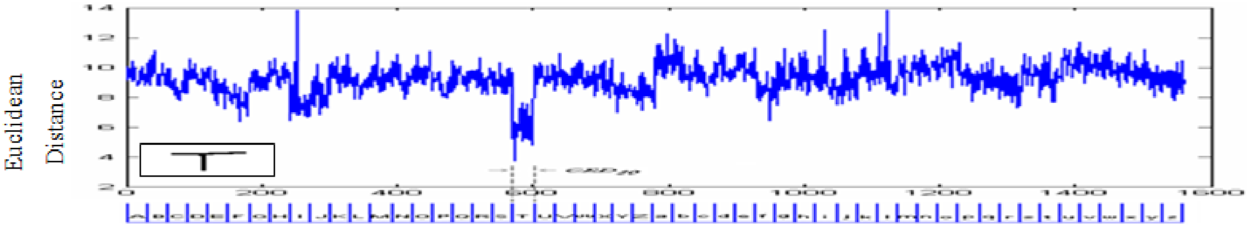


Figure 4. Letter "T" classification

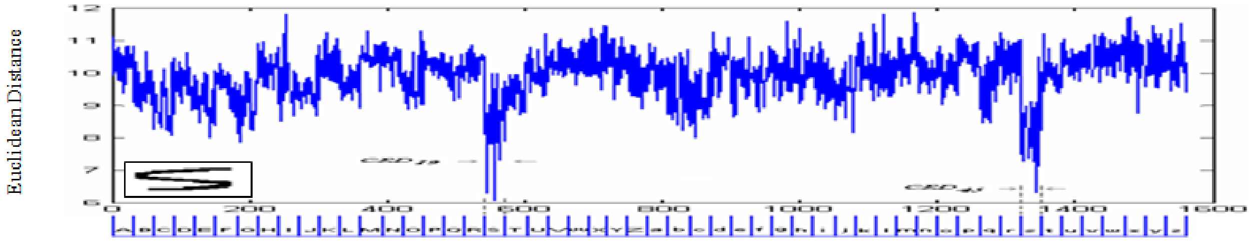
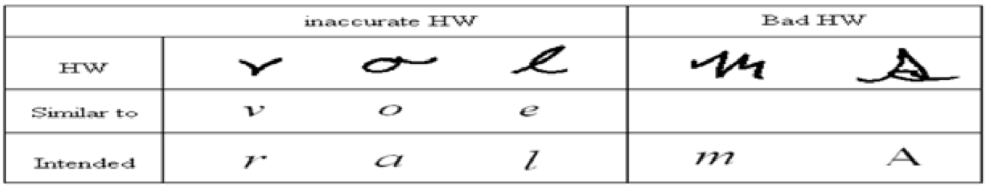


Figure 5. Letter "S" classification



	inaccurate HW			Bad HW	
HW					
Similar to	v	o	e		
Intended	r	a	l	m	A

Figure 6. Inaccurate and bad HW

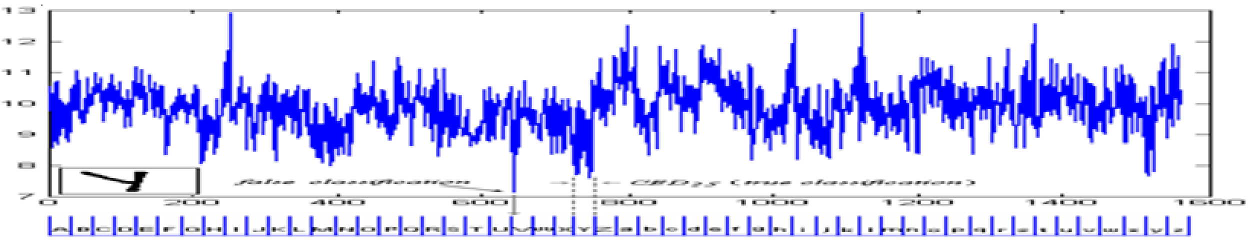


Figure 7. True and false classification

Sometimes the false classification could not be avoided due to the great similarity between the input HW letter and the unintended character class as shown in Figure 8. The input HW letter is " *r* ", but it is classified as " v " due to the great similarity between them. The solution of this problem is beyond the scope of this work (out of the proposed system ability). The Recognition process will be character by character and the designed program preserves the recognized characters to their words.

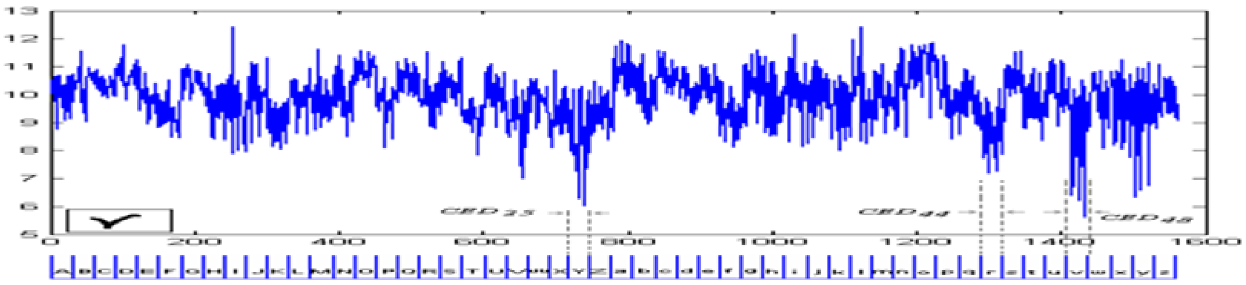


Figure 8. false classification due to not good HW

4.2. Post processing

The postprocessing step includes all processes that may be made to enhance the recognition and make the decision by different ways such as the prior contextual knowledge, integration of grammatical and syntactic knowledge, spelling, punctuation mark … etc. It focuses on solving the problem of the bad recognition between the characters between their upper and lower cases. The principle of its processing depends upon the comparison between the recognized characters within the same word. The normal word may contain all letters with their lowercases as "*university*" or may have only the first one with its uppercase especially if this word at the beginning of the sentence as "*University of Babylon*" or it represents a name as or "*Babylon*". It is not proper that the word is written in lowercase letters except one or more letters at its middle or last in uppercase as in "*uniVersitY*". Sometimes, the terms may be written with all letters in uppercases as in "*Handwriting Recognition*". This case can be easily discovered by the inspection of the recognized letters with the same word and the adjacent ones.

4.3. Data Base Collection and Document Image Acquisition

The proposed handwriting recognition system is with two phases; the first is the training phase that uses training handwritten samples while the second is the testing and recognition phase that needs test samples. The collected database for training must include different handwritten styles related to the scope of the proposed system. The data base was collected locally from various right hand and left hand writers with different ages, educations, temper etc. Characters were written by writers using specific forms on plain, white paper sheets with black ink pen to give clear strokes with sharp edges. Each filled form contains 52 Latin characters including the upper (A-Z), lower (a-z) cases and (0-9) numerical digits as shown in Figure 9. The collected data base depends upon the variety not on the quantity. The selection of the training samples must avoid as possible the redundant HW styles. The proposed handwriting recognition system processes data that were captured from a flatbed scanner. They were scanned at *150-dots-per-inch* resolution, in 256 levels of gray to produce one file per writer. The next task is to segment each form into its component characters. The pixel histogram calculations based segmentation algorithm takes a simple approach, looking for the gaps between lines and characters. Figure 10 shows the segmentation results.

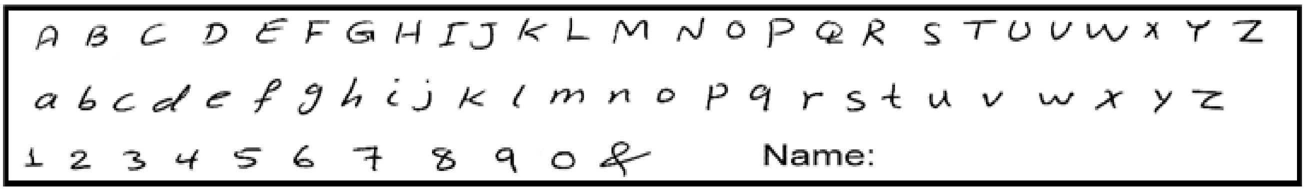

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
a b c d e f g h i j k l m n o p q r s t u v w x y z
1 2 3 4 5 6 7 8 9 0 &
Name:

Figure 9. HW data collection form

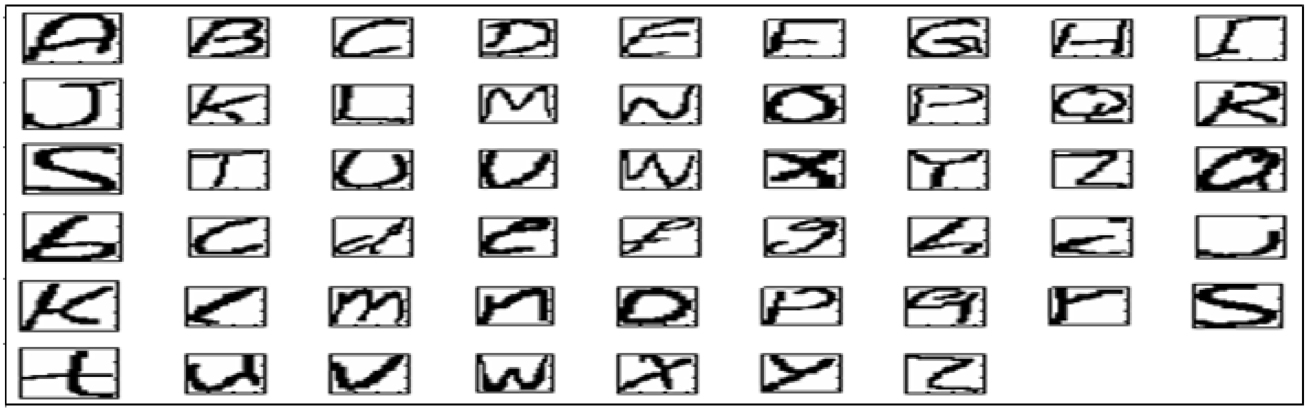


Figure 10. HW characters after the segmentation process from the form

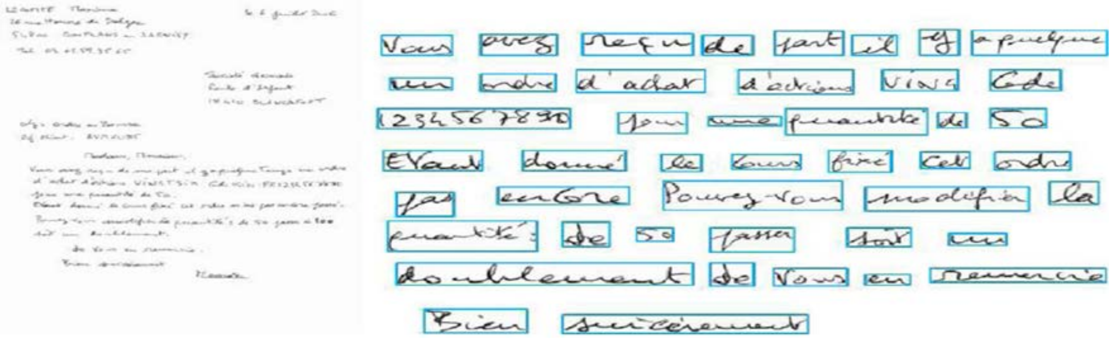

Vous avez reçu de part il y a quelque
un ordre d'achat d'adresse Vins Code
1234567890 Jour une quantité de 50
Etant donné le cours fixe cet ordre
pas encore Pouvez-vous modifier la
quantité de 50 passer tout ce
doublement de vous en remercie
Bien sincèrement

Figure 11. some mail letters handwritten of the Rimes of the training database and extractedwords data set classified as their classes and processed by the normalization steps

Each segmented character will pass through some steps (preprocessing steps) to be under the effects of same steps that the test samples will pass. These steps are binarization, thinning, thickening the strokes in order to smooth them and make all strokes approximately with the same thickness and characters resizing (size normalization) to be (32x32) pixels. These steps will make processing independent of strokes thickness and characters sizes. Figure 11 shows some of the characters of the training data set classified as their classes processed by the above steps. The attention during the data base collection was upon the quality not on quantity of the collected HW samples. The selection of the training samples process was avoiding the redundancy of characters samples. The redundancy is useless or unavailing. A large number of training HW samples may include bad HW writing styles and may make a negativity recognition process. Figure 12 shows the steps of the proposed HWRS. Figure 13 shows the block diagram of a complete proposed Handwriting Recognition System.

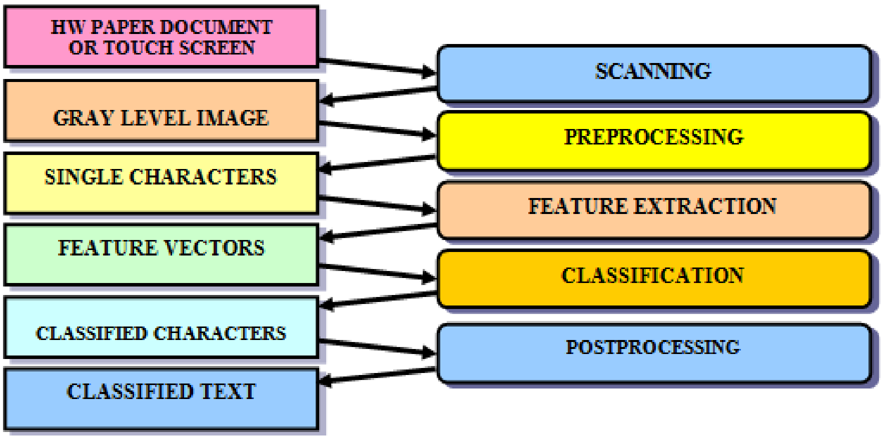


Figure 12. Steps of the proposed HWRS

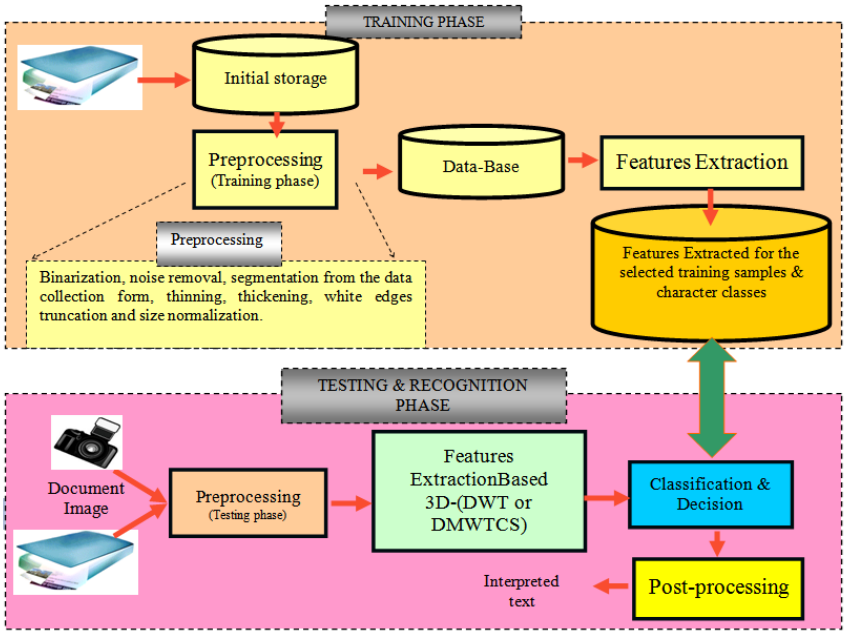

TRAINING PHASE
Initial storage
Preprocessing (Training phase)
Data-Base
Features Extraction
Preprocessing
Binarization, noise removal, segmentation from the data collection form, thinning, thickening, white edges truncation and size normalization.
Features Extracted for the selected training samples & character classes
TESTING & RECOGNITION PHASE
Document Image
Preprocessing (Testing phase)
Features ExtractionBased 3D-(DWT or DMWTCS)
Classification & Decision
Post-processing
Interpreted text

Figure 13. Block diagrams of the proposed HWRS

Figure 14 shows the raw handwritten texts written by different writers that under test in column (a) and the same texts after the recognition in column (b) by the proposed Handwriting Recognition System before the postprocessing step.

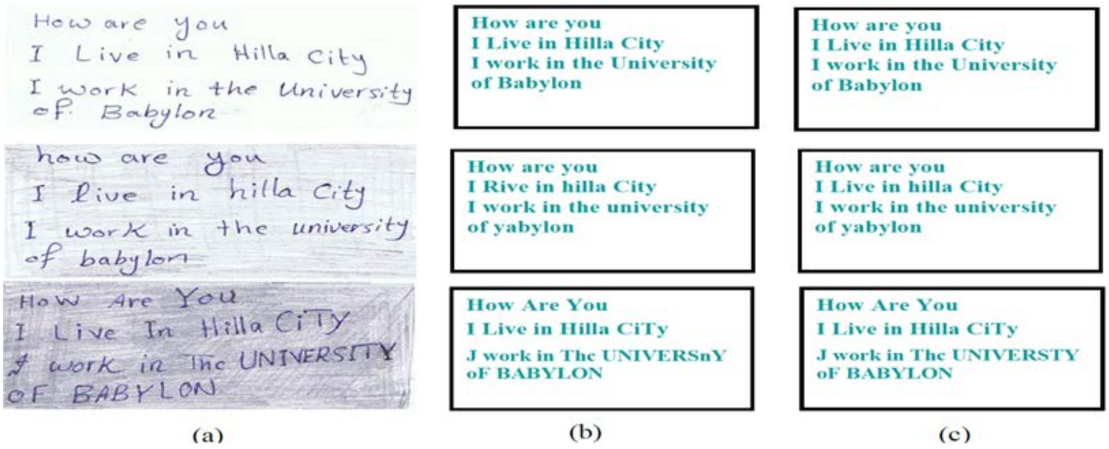

How are you
I Live in Hilla City
I work in the University
of Babylon
How are you
I Rive in hilla City
I work in the university
of yabylon
How are you
I Live in hilla City
I work in the university
of yabylon
How Are You
I Live in Hilla CiTy
J work in The UNIVERSnY
oF BABYLON
How Are You
I Live in Hilla CiTy
J work in The UNIVERSTY
oF BABYLON
(a)
(b)
(c)

Figure 14. HW Recognition examples (a) Raw Handwritten text. (b) The recognized text Using 2D-DMWTCS (c) The recognized text Using 3D-DMWTCS

5. Performance Evaluation of the Proposed HWRS

The Rimes database [26,27] was presented in 2006 and gathers more than 12,500 handwritten documents written by about 1,300 volunteers. It was created to provide data relative to advanced mailrooms, and the panel of documents offers large variability and makes the database a challenging one. Based on the Rimes database, it is possible to proceed logo recognition, document structure retrieval, and word and character recognition. Since 2007, evaluation campaigns have occurred [19] which enable participants to compare their results. In our work, we use the presegmentedword images given by the database to perform isolatedword recognition. The database consists of 59,203 wordimages divided into three subsets: 44,197 images for training, 7,542 for validation, and 7,464 for testing. The first experiment considers the full dictionary of size 5,334, including all words from the training, validation, and test sets. The second experiment considers only the reduced test dictionary of size 1,612 words. The last experiment uses the 4,943-words dictionary of training and validation databases, giving 392 out-of-vocabulary words (5.5 percent of the testset). About 300 new trigraphs are present in the test set and not in the other sets. The proposed handwriting recognition system is designed to be on-line character-based recognition system. Therefore; the system performance evaluation will be concerned with character recognition. This section focuses on the evaluation of the recognition task, accuracy rate, and the recognition time as well as the proposed integrated recognition system. The following evaluation experiments were made using a Laptop Toshiba of 2.2 GHz,4 dual cores, 8Gbyte internal memory RAM and the system has 2Mbyte cash memory. The proposed system was built and tested using Matlab2014a. The histograms and tables of this evaluation were achieved by using the locally collected HW samples.

5.1. Recognition Rate Evaluation

In order to evaluate the proposed system, two types of experiments were performed. In the first one the system was trained with 1560 training characters of 30 different HW styles. The Recognition Rate (RR %) was inspected by letters of the Latin alphabetic characters written by 100 writers. The inspection was for the uppercase (A-Z) and the lowercase (a-z) separately as shown in Figure 15 and Figure 16. It is clear that the characters having simple patterns like C, O can be recognized more accurately (high RR) than characters with more complexity (having multiple strokes and junctions) like R, K, q etc. By comparing the results shown in Figure 15 and Figure 16, the RR of the uppercase characters is higher than of their lowercases. The reason behind this result is that the uppercase letters are always more obvious than the lowercases since the first have more right-angled straight strokes than the curved ones.

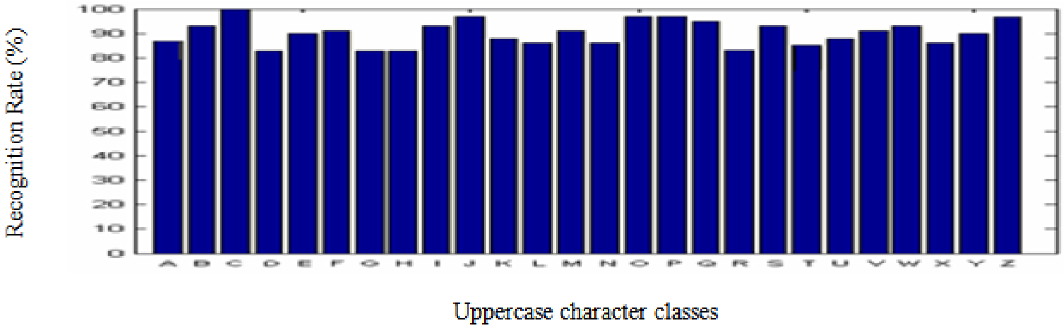


Figure 15. The Recognition Rate (%) inspected by letters of 100 writers for uppercase letters

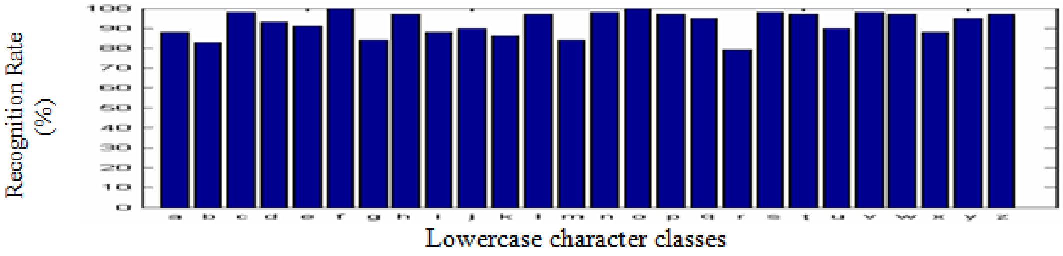


Figure 16. The Recognition Rate (%) inspected by letters of 100 writers for lowercase letters

The second experiment was the Recognition Rate (%) inspection for 100 testing samples for both upper and lower characters cases. Two of the test handwritten samples were the same as used for the training and they showed a RR of 100% (circled ones). Other test samples were written by the same persons who contributed in the training samples and these samples showed very high RR.

The other samples with neat, very clear and fine handwriting. The other test samples were new HW samples. The weak or bad HW test samples gave the worst results as shown in Figure 17. Table 1 summarizes the results of the two inspection experiments for 3D-Multiwavelet [17].

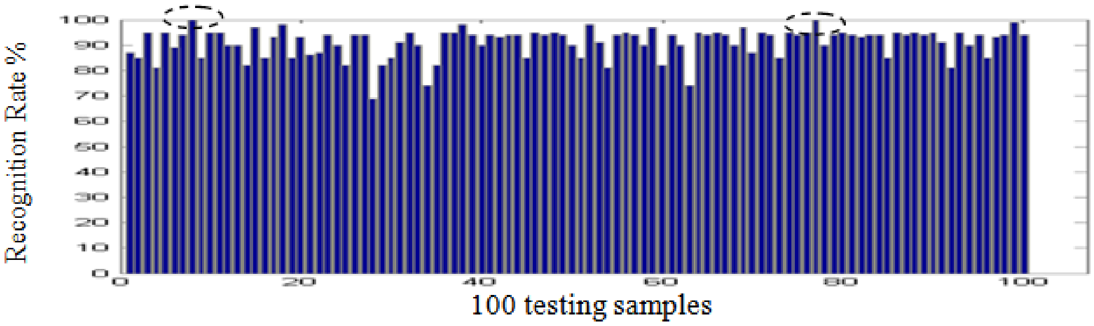


Figure 17. the Recognition Rate (%) inspected for 100 testing samples for both upper and lower characters cases

Table 1. Recognition Rate (%) of the proposed system for experiments 1 & 2

Characters Classes	2D-DMWTCS RR%	3D-DMWTCS RR%
Uppercase (26 classes) inspected by letters	94.28	96.94
Lowercase (26 classes) inspected by letters	91.63	93.67
Upper/lower (52 classes) inspected by writers	92.995	95.305

5.2. Recognition Time Computations

The main challenge is to speedup the recognition process and to improve the recognition accuracy. However, these two aspects are in mutual conflict. It is relatively easy to improve recognition speed while trading away some accuracy. But it is much harder to improve the recognition speed while preserving the accuracy. The recognition time of the proposed handwriting recognition system depends on many factors, The features and the specifications of the processing system like microprocessor speed and the available processing memory; size of the HW document to be recognized (number of sentences, words and characters); The scanning resolution (dpi); The degree of the HW document noise; The degree of the lines skew; degree of the characters slant; HW style (discrete or mixed styles), which affect on characters segmentation step; type and the decomposition level of the 3D-DMWT; number of the training data samples (size of the data-base); It is worth mentioning that the designed system uses some Matlab functions and subroutines that may slow down the processing speed. The goal is the minimization of the recognition time as possible. The recognition time that will be computed is per one character since the proposed HWRS is a character-based recognition. The HW document that is under test was a moderate noisy document and consists of six sentences (lines) with different skews namely L1, L2 … L6. These lines contain about 28 individual words (wo) with about 131 characters with different slants. The HW doc was scanned by flat scanner at 256 gray scales with 150-dpi resolution. The 3D-DMWTCS was by using Daubechies family at level 1 of decomposition that extracts 1024 features per character. Concerning the skew estimation and correction step, it is clear that the time required for long sentences is more than that for shorter ones. For the word segmentation, the time required to segment a line into individual words depends upon the number of the words and the characters which form it (line size). The time needed for the slant estimation and correction also depends upon the word size and its slant degree. As regard the thinning and thickening step, many experiments were made to compute the time required for this process and find the average time per character. There is interference between the segmentation and the recognition steps of the proposed system. Once the character is segmented, it will be delivered to the recognition step that includes the features extraction and the classification steps. Therefore; the needed time will be counted for these steps together.

6. General Discussion

The primary goal of the work is to design a complete modular on-line recognition system (writer-independent) for the handwritten text. This system deals with discrete and mixed HW styles in the upper and lower cases of Latin script letters and it is based on character recognition. The proposed system is a 3D-Discrete Multiwavelet Transform-based as a features extractor. The classifier was trained by a locally collected data base during the training phase. Patterns matching and classification is during the recognition phase. However, in every case the presented algorithm significantly improves the word and character segmentation and then recognition enhancement. With the application of the proposed approach one avoids any effect on the characters connectivity in the word and makes a negligible change in their aspect ratios and on shape nature. Some problems may arise when a word itself includes characters with different slants but rarely the writer himself writes a word with different characters orientations. However, in every case the presented algorithm significantly improves the character segmentation and then recognition enhancement. The reported results can, however, be considerably improved by training the system for a specific writer. Since the extracted rules are based on vectors consisting of human comprehensible information rather than opaque numerical data, a further improvement on the reported accuracy by refining the automatically extracted rules manually is possible.

7. Conclusions

The work developed in this paper aims to design an online system of high recognition accuracy of Latin handwritten types using (3D-DMWTCS). Multiwavelet representation has the advantage that the variations in the character shapes caused by the writing styles of different persons will cause only minor changes in the wavelet representation. The main information and features are concentrated at the approximation subband and the others are distributed in the other subband images. The recognition process depends on the features included in the approximation subband by about 53%. Using (3D-DMWTCS) for features extraction makes the recognition system need only a small training set to achieve high recognition accuracy regarding to the training samples (Data-Base), the collected data base depends upon the variety not on the quantity for the proposed HWRS based on (3D-DMWTCS). The selection of the training samples must avoid the redundant HW styles as possible. The relation between the number of the training samples and the Recognition Rate (RR) is not in direct proportional. The random increase in the training samples will not of necessity increase the RR; it may confuse the recognition process. Low resolution scanning (less than 100 dpi) will give erosion characters patterns. The high resolution scanning (more than 200 dpi) will increase the pixel size and improve the recognition but increase the time of the processing steps. The suitable resolution was 150 dpi. The capturing color mode has an effect on the recognition. The experiments show that the scanning with 256 gray scales is better than black/white or color mode. The proposed HWRS based on (3D-DMWTCS) system achieve an overall classification accuracy of 95.305 percent with 3D-DMWTCS using the Rimes database. These results have a gain of 2.31dB for 3D-DMWTCS compare with (2D-DMWTCS).

References

[1] Cun-Zhao Shi; Chun-Heng Wang; Bai-Hua Xiao; Song Gao; Jin-Long Hu, "Scene Text Recognition Using Structure-Guided Character Detection and Linguistic Knowledge," Circuits and Systems for Video Technology, IEEE Transactions on , vol.24, no.7, pp.1235,1250, July 2014

[2] A.-L. Bianne, C. Kermorvant, and L. Likforman-Sulem, “Context-Dependent HMM Modeling Using Tree-Based Clustering for the Recognition of Handwritten Words,” Proc. SPIE Document Recognition and Retrieval, 2010..

[3] Assabie Y. and Bigun J" Writer-independent offline recognition of handwritten Ethiopic characters", In: Proc. 11th ICFHR, August 19-21, Montreal, 2008.

[4] Ballesteros J., Travieso C.M., Alonso J.B. and Ferrer M.A, “ Slant estimation of handwritten characters by means of Zernike moments”, IEE electronics letters vol.41 no.20, 29th September,2005

[5] Bruce K., “Multiwavelets for Quantitative Pattern Matching” Proceedings of the 42nd Hawaii International Conference on System Sciences, pp 1- 10,2009.

[6] Cheng-Lin L., Jaeger S. and Nakagawa M. “Online recognition of Chinese characters: the state-of-the-art”, IEEE transactions on pattern analysis and machine intelligence, vol. 26, no. 2, February 2004.

[7] Gao X., Fen X., and Bodong L., “Construction of Arbitrary Dimensional Biorthogonal Multiwavelet Using Lifting Scheme” IEEE Transactions On Image Processing, VOL. 18, NO. 5, MAY, pp 942-955,2009.

[8] Deselaers, T.; Keysers, D.; Hosang, J.; Rowley, H.A., "GyroPen: Gyroscopes for Pen-Input With Mobile Phones," Human-Machine Systems, IEEE Transactions on , vol.45, no.2, pp.263,271, April 2015.

[9] Shahriarpour, E.; Sadri, J., "Recognition of legal amount words on Persian bank checks using Hidden Markov Model," Intelligent Systems (ICIS), 2014 Iranian Conference on , vol., no., pp.1,5, 4-6 Feb. 2014

[10] Breuel, T.M.; Ul-Hasan, A.; Al-Azawi, M.A.; Shafait, F., "High-Performance OCR for Printed English and Fraktur Using LSTM Networks," Document Analysis and Recognition (ICDAR), 2013 12th International Conference on , vol., no., pp.683,687, 25-28 Aug. 2013

[11] El-Anzy L., " Design and Simulation of STBC-(OFDM and CDMA) Transceivers based on Hybrid Transforms ", Ph.D. Thesis, Univ. of Technology, Electrical and electronic engineering, Dep., Oct.2006.

[12] Rodolfo P., Gabriela S., Tsang I. and George D “Text Line Segmentation Based on Morphology and Histogram Projection” 10th International Conference on Document Analysis and Recognition, pp 651-655., 2009.

[13] Suriya K. , Kitti A., Thanatchai K. “Recognition of power quality events by using Multiwavelet-based neural networks” Elsevier Ltd., Electrical Power and Energy Systems Vol 30 ,pp 254–260, 2008.

[14] K. Ubul, A. Adler and M. Yasin, “Multi-Stage Based Feature Extraction Methods forUyghur Handwriting Based Writer Identification,” In Genetic Algorithms in Applications, InTech , 2012.

[15] Shanker A., Tajagopalan A. “Off-line signature verification using DTW”, Pattern Recognition Lett. 28, pp1407–1414, 2007.

[16] Xiangqian Wu; Youbao Tang; Wei Bu, "Offline Text-Independent Writer Identification Based on Scale Invariant Feature Transform," Information Forensics and Security, IEEE Transactions on , vol.9, no.3, pp.526,536, March 2014

[17] Yousri K., Thierry P., AbdelMajid B., “A Multi-Lingual Recognition System for Arabic and Latin Handwriting” 10th International Conference on Document Analysis and Recognition, pp 1196-1200, 2009.

[18] Matheel E., Rabab F., “Novel Video Denoising Using 3-D Transformation Techniques,” International Journal of Engineering and Advanced Technology (IJEAT), Volume-2, Issue-5, June 2013.

[19] R. Cruz, G. Cavalcanti, and T. Ren, “An ensemble classifier for offline cursive character recognition using multiple feature extraction techniques,” in The 2010 International Joint Conference on Neural Networks (IJCNN),pp. 1 –8, July 2010.

[20] AbuzaraidaM. A., A. M. Zeki and A. M. Zeki, "Segmentation Techniques for Online Arabic Handwriting Recognition: Asurvey," in 3rd International Conference on Information and Communication Technology for the Moslem World, Jakarta,Indonesia, , pp. D37-D40,2010.

[21] R. Smith, “History of the Tesseract OCR engine: what worked and what didn’t ,” in DRR XX, San Francisco, USA, Feb. 2013.

[22] KN. Sriraam and R. Shyamsunder. “3-D Medical Image Compression Using 3-DWavelet Coders”, Digital Signal Processing, 21(1):100–109, January 2011.

[23] Ibrayim, M., and Askar H., “Design and implementation of prototype system for online handwritten Uyghur character recognition,” Wuhan University Journal of Natural Sciences,vol. 17 no. 2, pp. 131-136, April 2012.

[24] Dreuw, P., Rybach, D., Heigold, G., and Ney, H., Guide to OCR for Arabic Scripts Chp. Part II: RWTH OCR: A Large Vocabulary Optical Character Recognition System for Arabic Scripts, Springer, London, UK, pp. 215-254,July, 2012.

[25] Abuzaraida, M.A.; Zeki, A.M.; Zeki, A.M., "Feature extraction techniques of online handwriting arabic text recognition," Information and Communication Technology for the Muslim World (ICT4M), 2013 5th International Conference on , vol., no., pp.1,7, 26-27 March 2013

[26] http://rimes.it sudparis.eu/, 2010.

[27] E. Augustin, M. Carre, E. Grosicki, J.-M. Brodin, E. Geoffrois, and F. Preteux, “Rimes Evaluation Campaign for Handwritten Mail Processing,” Proc. Int’l Workshop Frontiers in Handwriting Recognition, pp. 231-235, 2006.