A Modified Back Propagation Algorithm for Assyrian Optical Character Recognition Based on Moments

Dr. Luma Fayeq Jalil
Computer Science Department, University of Technology/Baghdad.
Modhar Mohsen Mohammed
Computer Science Department, University of Technology/Baghdad.

Received on:7/6/2015 & Accepted on:17/12/2015

ABSTRACT

Character recognition has been very popular and interested area for researches, and it continues to be a challenging and impressive research topic due to its diverse applicable environment. The optical character recognition has been introduced as a fast and accurate method to convert both existing text images as well as large archives of existing paper documents to editable digital text format.

However, existing optical character recognition algorithms suffer from flawed tradeoffs between accuracy and speed, making them less effective and impractical for large and complex documents. This paper describes a suggested method for Assyrian optical character recognition using modified back propagation artificial neural network based on moments. The experimental results show that the proposed method achieves higher recognition accuracy rate in compared with the standard algorithm.

Keywords: OCR, Assyrian characters, Segmentation, Feature Extraction, Classification, Training, Back Propagation Neural Network Algorithm.

INTRODUCTION

Character recognition is an art of detecting, segmenting, and identifying characters from document image. More precisely Character recognition is process of detecting and recognizing characters from input document and converts it into an equivalent machine form. Character recognition is one of the most recognition systems and it is getting more and more attention since last decade due to its wide range of applications [1].

There are many important documents of history, such as manuscripts, that can be converted into machine editable file so that it can be easily accessed and processed. The objective of OCR software is to recognize the captured text and then convert it into digital text format. A document is first scanned by an optical scanner, which produces a non-editable image format, then computer algorithms are implemented to identify the characters in that document.

This paper presents the proposed approach as a multi-stages Assyrian character recognizer. The designed project consists of a number of pre-processing steps followed by the actual recognition using an artificial neural network learned by a modification of back-propagation algorithm based on predefined moments.

Each character of the input Assyrian text will be represented by a pattern of seven moment invariants computed for that character, then this pattern is applied to the modified back-propagation neural network to recognize the input characters efficiently and accurately.

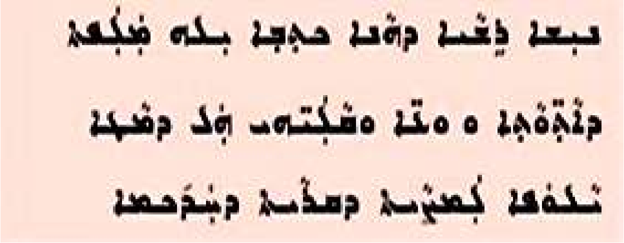
Moments

https://doi.org/10.30684/etj.34.2B.8
2412-0758/University of Technology-Iraq, Baghdad, Iraq

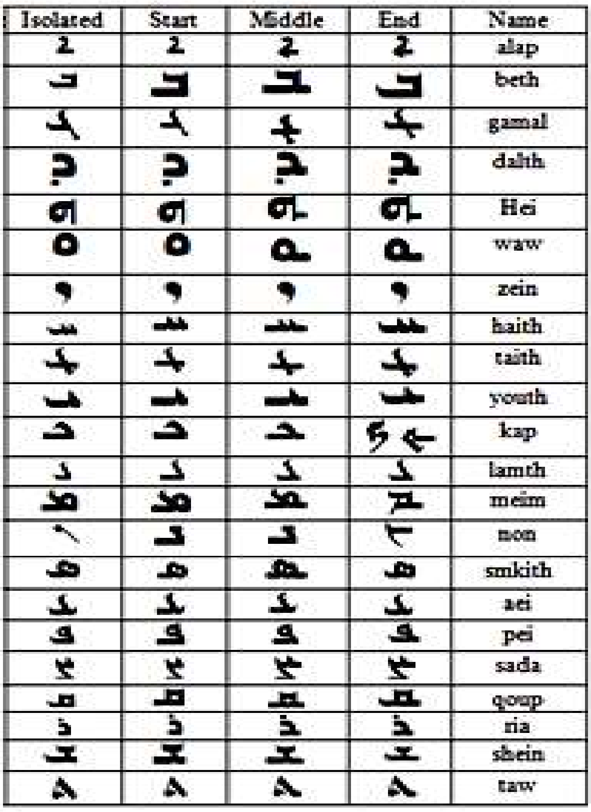
In 1961, Hu introduced moment invariants. Based on the theory of algebraic invariants he derived a set of moments that are position, size, and orientation independent. However, regular moments are not orthogonal and as a consequence, reconstructing the image from the moments is deemed to be a difficult task [2]. Moment based feature descriptors has been considered a powerful tool for image analysis applications. Geometric moments present a low computational cost, but are highly sensitive to noise [3].

The Characteristics Of Modern Assyrian Scripts

Assyrian also known as Syriac Aramaic is a dialect of Middle Aramaic that was once spoken across much of the Fertile Crescent and Eastern Arabia. Classical Syriac became a major literary language throughout the Middle East from the 4th to the 8th centuries; Syriac literature comprises roughly 90% of the extant Aramaic literature. The Aramaic language in history is linguistically the Middle Aramaic but since most Christian scribes of the Christian manuscripts, who wrote in the Middle Aramaic, lived in this region of Assyria, which had come to refer to both the historical Assyria as well as the Levant, this specific dialect of the Middle Aramaic has come to be known as Assyrian. Assyrian remains the liturgical language of Syriac Christianity to this day. Assyrian is a Middle Aramaic language, and, as such, it is a language of the Northwestern branch of the Semitic family. Before Arabic became the dominant language, Syriac was a major language among Christian communities in the Middle East, Central Asia and Kerala, and remains so among the Assyrians and Syriac-Arameans to this day. Figure (1) shows the Assyrian script and figure (2) shows Assyrian characters [4, 5, 6].



Figure(1): Assyrian Script.


Isolated
Start
Middle
End
Name
alap
beth
gamal
dalth
Hei
waw
zein
haith
taith
youth
kap
lamth
meim
non
smkith
aei
pei
sada
qoup
ria
shein
taw

Figure(2): The Assyrian Characters.

The Proposed Approach

A captured document is entered to designed approach as a BMP image type, which is the most widely used format that represents a real-world uncompressed image. The pixels of BMP images can be easily sorted in two dimensional array to be used for programming methods [7].

The input document image is treated by several processes in order to recognize the text within it, as described below:

Preprocessing

The document image that is gotten from scanners may acquire some amount of unwanted information and noise. The preprocessing transforms input images into more acceptable and regular form for the next OCR stages. The preprocessing stage consists of a series of processes, as follows [8, 9]:

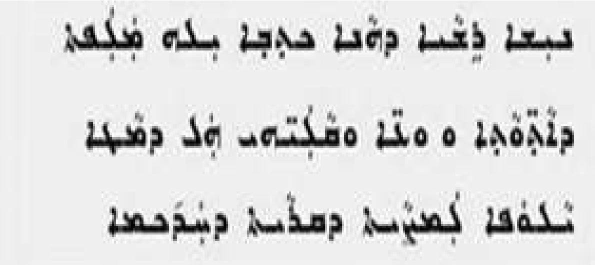
Grayscale Conversion Process:

It is the convert of the color document images with color depth of 24-bit into grayscale copies of 8-bit depth by averaging the three color values (RGB). The reduction of color depth is represented by the following method [10, 11].

For each pixel in document image (Pi_{xy})

Set $pi_{xy\,new} = (Pi_{xy}.R + Pi_{xy}.G + Pi_{xy}.B) / 3$ $\{0 \le x \le$ image width, $0 \le y \le$ image length$\}$

Where R is the Red color intensity, G is the Green color intensity, and B is the Blue color intensity of a pixel. The result of this method is shown in figure (3).

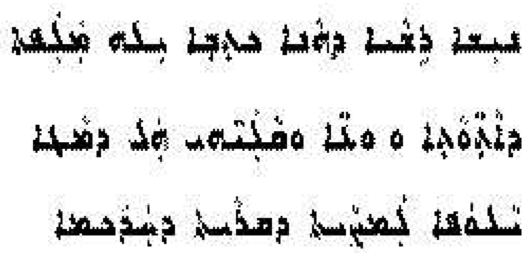


Figure(3): Grayscale Image.

Binarization Process:

It is the convert of grayscale images into copies of black and white only, which is called binary images. Binarization process separates images into two parts of pixels using a global or local threshold: the first part is called foreground or information that contains black pixels, and second part is called background that contains white pixels. The proposed approach makes use of the J. R. Parker iterative method to determine a global threshold as illustrated in the below pseudo code. The result of the binarization is shown in figure (4) [12, 13, 14, 15].

```
For each pixel $Pi_{xy}$   {$0 \le x \le$ image width, $0 \le y \le$ image length }
Threshold = $\sum_x \sum_y Pi_{xy} / (x * y)$
While
For each pixel $Pi_{xy}$ where $Pi_{xy} \le$ threshold
Set $\mu 1 = \sum_x \sum_y Pi_{xy}$ / No. of pixels $\le$ threshold
For each pixel $Pi_{xy}$ where $Pi_{xy} >$ threshold
Set $\mu 2 = \sum_x \sum_y Pi_{xy}$ / No. of pixels $>$ threshold
If  threshold = $(\mu 1 + \mu 2) / 2$  then  Exit while {The threshold has been found}
Else Set Threshold = $(\mu 1 + \mu 2) / 2$
   For each pixel $Pi_{xy}$    If $Pi_{xy} \le$ Threshold then Set $Pi_{xy\,new} = 0$ Else Set $p_{xy\,new} = 1$
```

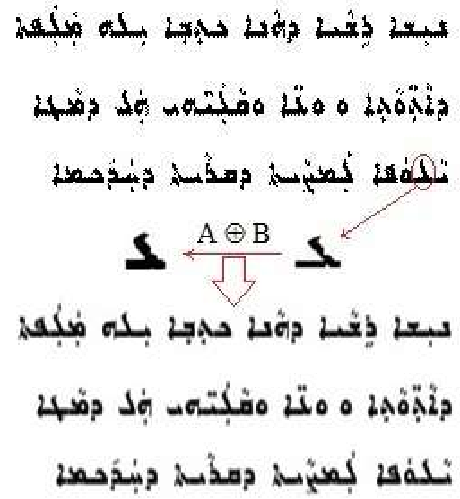


Figure(4): Binary Image.

Dilation Process:

Increases the thickness of objects or grows shapes in order to fill the small holes and bridge gaps by using a predefined structuring element. The dilation is denoted by A ⊕ B, where A is the processed image and B is the structuring element. Figure (5) shows the result of the dilation of a document image. The following pseudo code illustrates the implementation of this process [16, 17].

```
For each pixel $Pi_{xy}$ where $Pi_{xy} = 0$   {$0 \le x \le$ image width, $0 \le y \le$ image length }
If any 8-neighbors pixel of the $Pi_{xy}$ equal *one* Then
 Set $Pi_{xy}$ to one
```

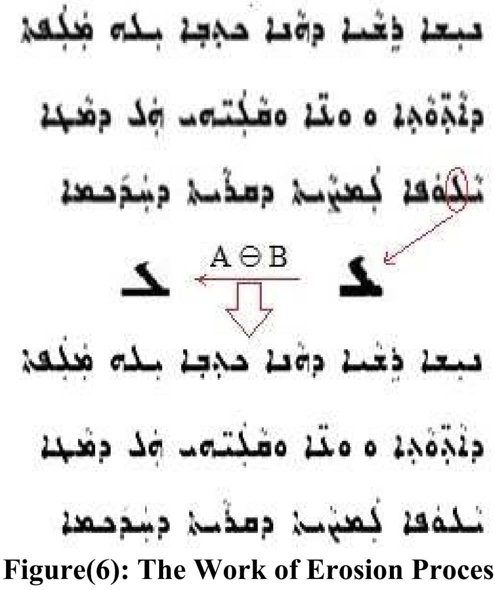


Figure(5): The Work of Dilation Process.

Erosion Process:

Thins or shrinks the objects in order to eliminate irrelevant detail by using a predefined structuring element. The erosion is denoted by A ⊖ B, where A is the processed image and B is the structuring element. Figure (6) shows the result of the erosion of a document image. The following pseudo code illustrates the implementation of this process [16, 17].

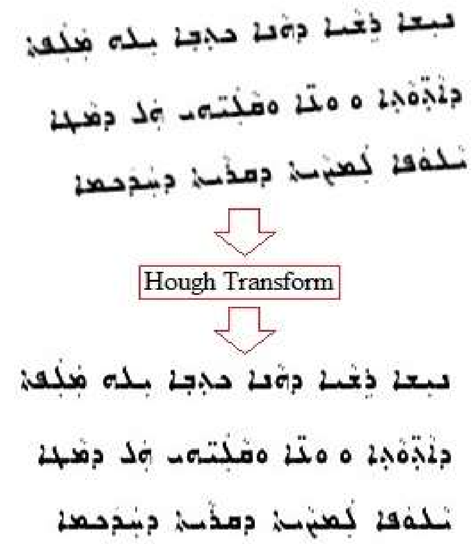
```
For each pixel $Pi_{xy}$ where $Pi_{xy}$ = 1
{0 ≤$x$≤ image width, 0 ≤$y$≤ image length }
 If any 8-neighbors pixel of the $Pi_{xy}$ equal zeroThen
 Set $Pi_{xy}$ to zero
```



Figure(6): The Work of Erosion Process.

Skew Correction Process:

Capturing a document by digital cameras or scanners may cause to skew the text of that document image which in turn will lead to bad segmentation and recognition. Therefore, the correction of skewed text lines is needed. The proposed approach determines the skew angle of the text lines by the Hough Transform in which the straight line is represented as $y = mx + b$, where the pair x and y is a point within image, m is a slope parameter, and b is a intercept parameter [18, 19, 20]. Figure (7) shows a skewed text of a document image and its correction.



Figure(7): Skewed Text and Its Correction.

Thinning Process:

As the name implies, this process produces the object skeletons of one pixel. The proposed approach uses the ***Zhang-Suen*** thinning algorithm, which has two phases [21, 22, 23]:

Phase one: the pixels of value (one) will be set to (zero) if and only if the following conditions are satisfied:

a1) $2 \leq N(P_{x,y}) \leq 6$

b1) $S(P_{x,y}) = 1$

c1) $P_{x,y-1} * P_{x+1,y} * P_{x,y+1} = 0$

d1) $P_{x+1,y} * P_{x,y+1} * P_{x-1,y} = 0$

Phase two: the pixels of value (one) will be set to (zero) if and only if the following conditions are satisfied:

a2) $2 \leq N(P_{x,y}) \leq 6$

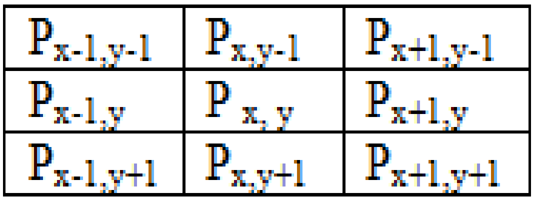
b2) $S(P_{x,y}) = 1$

c2) $P_{x,y-1} * P_{x+1,y} * P_{x-1,y} = 0$

d2) $P_{x,y-1} * P_{x,y+1} * P_{x-1,y} = 0$

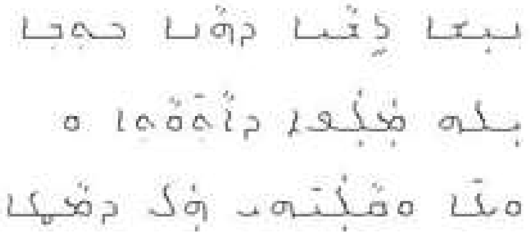
Where $N(P_{x,y})$ is the number of neighbors have value one, $S(P_{x,y})$ is the number of clockwise transition from 0 to 1. Figure (8) explain the eight neighbors arrangement, figure (9) shows the results of the thinning process, and the below pseudo code illustrates the implementation of the thinning process.

```
Set foreground to 1, background to 0
While
Set Image_new equal to image_old
For each pixel $Pi_{xy}$ of image_old where $Pi_{xy} = 1$ {$0 \leq x \leq$ image width, $0 \leq y \leq$ image length }
If $2 \leq$ (No. of neighbors of($Pi_{xy}$)equal to one) $\leq 6$ then
 If (No. of neighbors transition from 0 to 1 (clockwise) of $Pi_{xy}$) =1 then
If $Pi_{x,y-1} * Pi_{x+1,y} * Pi_{x,y+1} = 0$ then
If $Pi_{x+1,y} * Pi_{x,y+1} * Pi_{x-1,y} = 0$ then
Set $Pi_{xy}$ of image_new equal to zero
Set image_old equal to Image_new
For each pixel $Pi_{xy}$ of image_old where $Pi_{xy} = 1$
If $2 \leq$ (No. of neighbors of($Pi_{xy}$)equal to one) $\leq 6$ then
If (No. of neighbors transition from 0 to 1 (clockwise) of $Pi_{xy}$) =1 then
If $Pi_{x,y-1} * Pi_{x+1,y} * Pi_{x-1,y} = 0$ then
If $Pi_{x,y-1} * Pi_{x,y+1} * Pi_{x-1,y} = 0$ then
Set $Pi_{xy}$ of $\text{image}_{new}$ equal to zero
Set $\text{image}_{old}$ equal to $\text{Image}_{new}$
If No change in any pixel then exit while
```



$P_{x-1,y-1}$	$P_{x,y-1}$	$P_{x+1,y-1}$
$P_{x-1,y}$	$P_{x,y}$	$P_{x+1,y}$
$P_{x-1,y+1}$	$P_{x,y+1}$	$P_{x+1,y+1}$

Figure(8): The Arrangement of Neighbor Pixels.



Figure(9): The Thinning Process Result.

Segmentation

Segmentation is the technique that used to determine and isolate text lines, words, characters, associated dots, and accents. It is easy to implement, but difficulties comes when segmenting cursive language, like the Arabic language, the Assyrian languages, etc. The more successful segmentation leads to higher recognition accuracy. The proposed approach uses the projection profile technique detect the text lines, words and characters [24, 25].

Lines Segmentation:

The horizontal projection profile has been used to detect the lines of the input text. The white pixels that come along a row indicate the line ends [24, 25, 26]. Figure (10) shows the line detection. The following pseudo code describes the use of horizontal projection profile for line segmentation.

```
For each pixel $Pi_{xy}$ {$0 \le x \le$ image width, $0 \le y \le$ image length}
If any pixel in line $y = 0$ then
For each pixel $Pi_{x,y+1}$
If all pixels in line $y+1 = 1$ then
Add newline
```

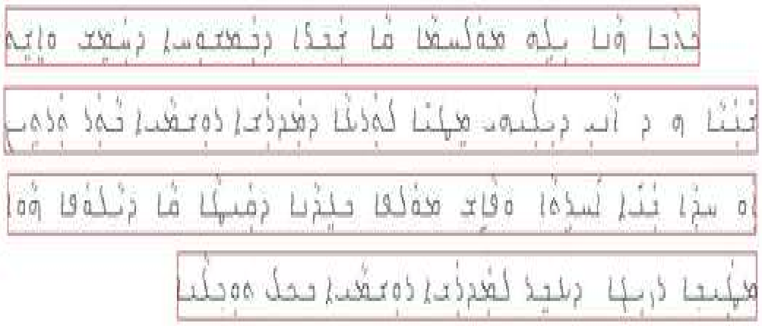
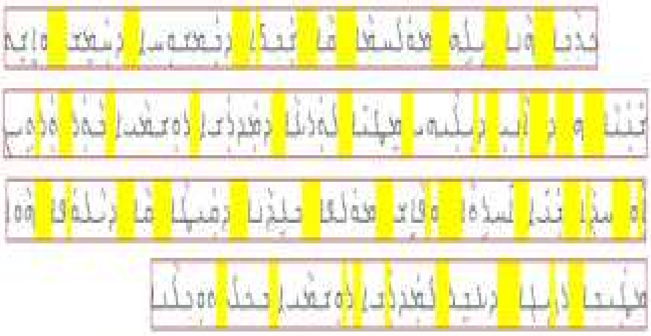


Figure (10): The Line Detection.

Words Segmentation:

The vertical projection profile has been used to detect the words of the input lines. The white pixels that come along a column of a line indicate the word ends [25, 26]. Figure (11) shows the word detection. The following pseudo code describes the word segmentation by vertical projection profile.

```
For each line segment
Word_end = false
For each $x$ {0 ≤$x$≤ line length}
For each $y$ {0 ≤$y$≤ line width}
If all pixels $Pi_{xy}$ = 1 then
Word_end = true
Else
If Word_end = true then
Add new_word
Word_end = false
```



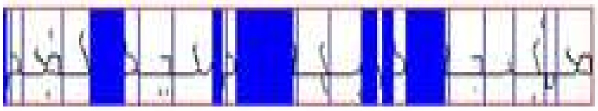
Figure(11): The Word Detection.

Characters Segmentation:

The vertical projection profile has been used to detect the characters of the input words. The rapid change of black pixels from single occurrence to multiple occurrences indicates the character ends [25, 26]. Figure (12) shows the character detection. The following pseudo code describes the use of vertical projection profile for character segmentation.

```
For each Word segment
Char_end = false
For each $x$ {Word length ≥$x$≥ 0 }
For each $y$ {0 ≤$y$≤ line width}
Increment ( No. of pixels $Pi_{xy}$ equal zero) by 1 where $Pi_{xy}$ = 0
If ( No. of pixels $Pi_{xy}$ equal zero) = 1 then
```

```
Char_end =  true
Else  If    Char_end = true  then Add    new_Character Char_end = false
```



Figure(12): The Character Detection.

Features extraction

Feature extraction is the technique that allows to capture the global character shape information and puts them in vectors which represent the identities of these characters. The proposed approach uses Moments method to extract character features that are invariant to scale, translation, rotation and reflection because they represent the measure of the pixel distribution around the centre of gravity of the characters in process. The method produces seven invariants of moments computed from the central character moment through order three [27, 28 ,29]. The regular moment is:

$m_{pq} = \sum_x \sum_y x^p y^q \; f(x,y)$

Where (M_{pq}) parameter is the two-dimensional moments of image function [$f(x, y)$]. The index of the moment invariants is ($p + q$), where (p) and (q) are both natural numbers.

The central moments can be calculated by:

$\mu_{pq} = \sum_x \sum_y (x - x^c)p\ (y - y^c)q\ \ f(x, y)$

Where (x^c) and (y^c) are the centre coordinates of gravity of character, and are calculated by:

$x^c = m_{10} \ / \ m_{00} \quad , y^c = m_{01} \ / \ m_{00}$

The moments normalized calculated by:

$\eta_{pq} \ = \ \mu_{pq} \ / \ \mu_{00}{}^{\gamma}$

Where the normalization factor is computed by:

$\gamma = (p+ q/2) +1$

The seven moment invariants calculated by:

$\Phi_1 = \eta_{20} + \eta_{02}$

$\Phi_2 = (\eta_{20} - \eta_{02})^2 + 4\ \eta_{11}{}^2$

$\Phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2$

$\Phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{03} + \eta_{21})^2$

$\Phi_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})^2 [(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03}) \times [3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$

$\Phi_6 = (\eta_{20} - \eta_{02}) [(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})$

$\Phi_7 = (3\eta_{21} - \eta_{03})^2 (\eta_{30} + \eta_{12}) [(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{12} - \eta_{30})^2 (\eta_{21} + \eta_{03}) \times [3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{30})^2]$

The following pseudo code describes the Moments method implementation.

```
For each Segmented Character
For each p { 0 ≤p ≤ 1}
For each q { 0 ≤q ≤ 1}
For each pixel $Pi_{xy}$ {0 ≤x ≤ character length , 0 ≤y ≤ character width}
$m_{pq} = \sum_x \sum_y x^p . y^q . Pi_{xy}$
$xc = m_{10} / m_{00}$
$yc = m_{01} / m_{00}$     {where xc, yc is central moments}
For each p  { 0 ≤p ≤ 3}
For each q   { 0 ≤q ≤ 3}
For each pixel $Pi_{xy}$  {0 ≤x ≤ character length , 0 ≤y ≤ character width}
$\mu_{pq} = \sum_x \sum_y (x - xc)^p . (y - yc)^q . Pi_{xy}$  {where $\mu_{pq}$ is discrete}
For each p       {0 ≤p ≤3}
For each $q$ { $0 \leq q \leq 3$}
$\gamma_{pq} = ((p + q) / 2) + 1$ { where $\gamma$ is normalization factor }
$\eta_{pq} = \mu_{pq} / \mu_{00}^{\gamma pq}$ {where $\eta$ is normalized central moments }
Compute the seven moment invariants
Next Segmented Character
```

Recognition

The proposed approach uses the Artificial Neural Network (ANN) to identify the Assyrian character images that are represented by vectors of seven invariant moments each.

An ANN is a computational model that consisted of simple elements arranged in a layer structure. These elements are called nodes (or neurons) and are operating in parallel in a connectionist approach of computation, i.e. the nodes are connected in a network from the input layer to the output layer and these connections are weighted. Usually an ANN calculates its output values by summing the weighted nodes and passes the summation to an activation function, like sigmoid function, to produce the results. The ANN is a system that considered to be adaptive because it changes its internal information (connection weights) according to the external information that flows through the network during the learning stage [29, 30, 31].

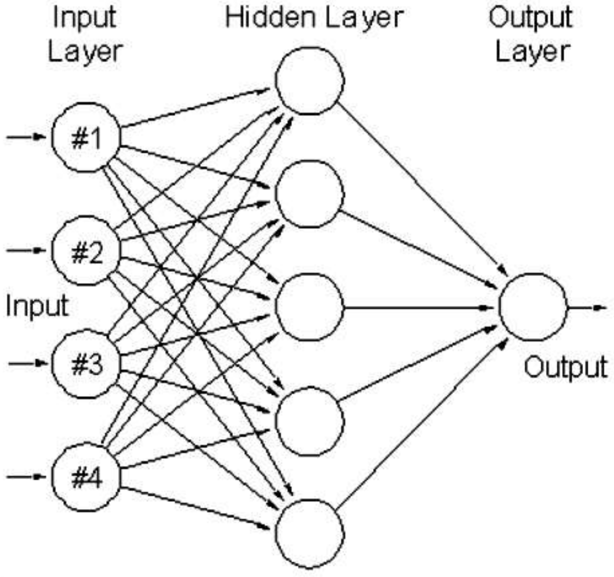
The learning of an ANN is the attempt of reducing the errors that occur between the target values and the output results. This stage can be done by the use of a machine learning technique like Back Propagation Neural Network (BPNN). The term BPNN used to describe feed-forward neural networks trained using the Back Propagation (BP) method. The architecture of BPNN is multilayer (Input, Hidden, and Output layers) and the nodes between adjacent layers are fully connected with variant weights, as shown in figure (13). The BP is a supervised learning method that has two phases [32,33,34]:

Forward Phase:

In which, the outputs that are calculated in one layer are feeds forward to the nodes in the next layer.

Backward Phase:

In which, the errors are computed by the use of an error function, like root mean square error, and if the result is not accepted then update the connection weights in the feed backward way to obtain new outputs in the next iteration. Otherwise, the ANN is considered to be learned



Figure(13): The Multilayer Neural Network.

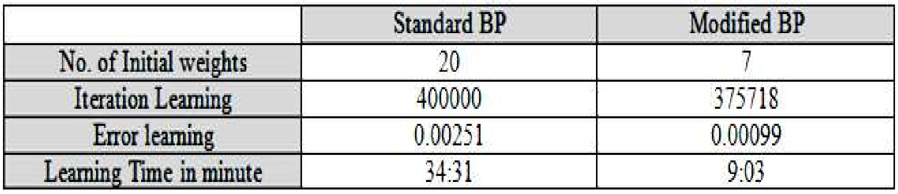
The proposed approach suggests a Modified Back Propagation Neural Network (MoBPNN), which differs than the standard BPNN with the following points:

1. The MoBPNN has three phases: Forward, Sort, and Backward phases. The Forward phase is similar to the standard BPNN, The Sort phase arranges the input vectors in ascending order according to their outputs, while the backward phase updates the connection weights using the biggest computed error only in order to approximate the output results to the Target values with less iterations.
2. The initial Target values of the MoBPNN are computed automatically by incrementing a specific value among them depending on the number of input vectors.
3. The MoBPNN is faster than the standard one, as shown in figure (14).
4. The MoBPNN has proved higher accuracy rate in optical character recognition, as shown in figure (15).
5. The MoBPNN is less CPU consuming than the standard BPNN.

The following pseudo code illustrates the Modified Back Propagation.

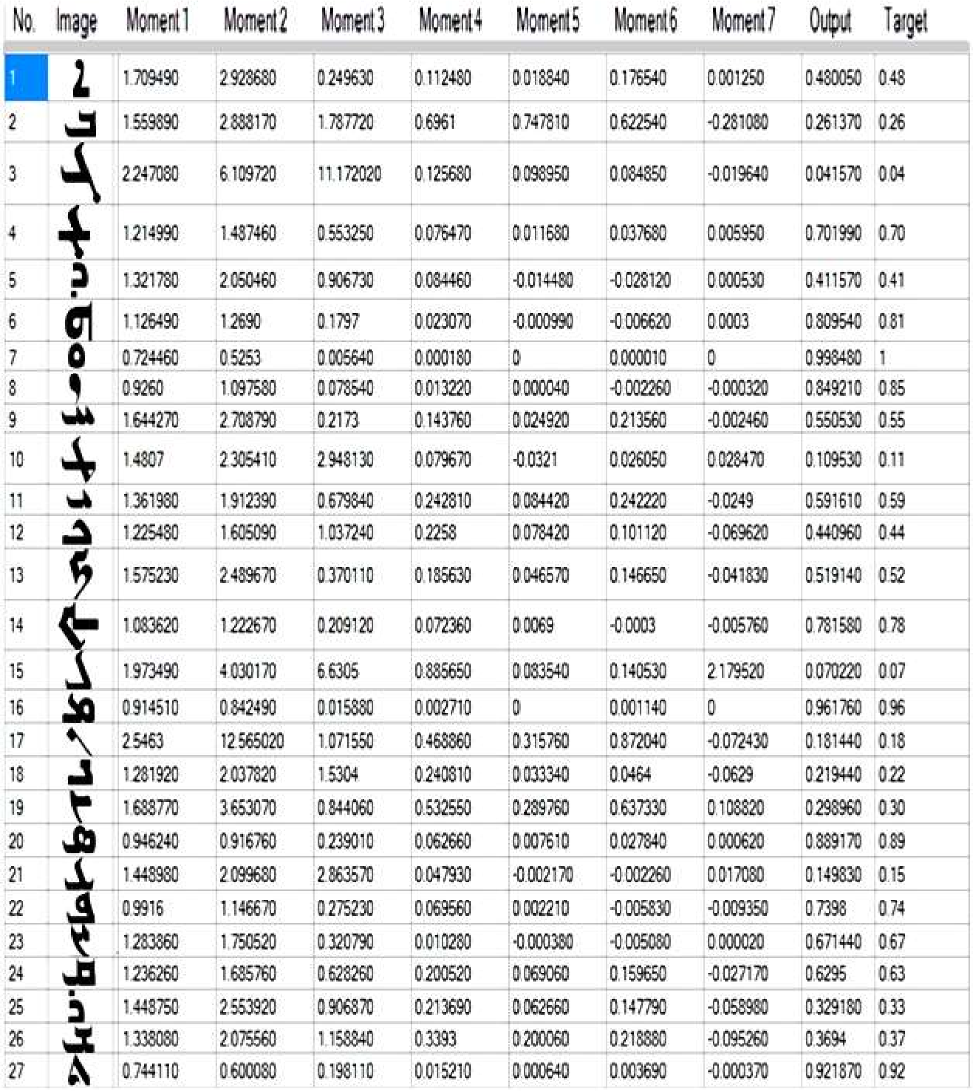
```
Set decimal_space to one
While not(Determine the values of  targets ($T_i$))
Decimal_space = decimal_space *  0.1
If ( decimal_space * vectors_no) equal or less than one
Difference_ among _ targets = 1 / vectors_no
For each target($T_i$)
$T_i$ =  Difference_ among _ targets * i
Set Error_learning to (decimal_space * 0.1)
Initialize all weights  in network randomly. {$V_{ij}$ for hidden layer, $W_i$ for Output layer}
Set learning-condition to false
Set ɳ to no.e.g.( ɳ = 1)              {learning rate}
Set iteration to zero
While Not (learning-condition)

#Forward Phase#
For each vector ($X_i$)
For each Hidden Layer unit ($H_j$)
For each Input unit ($I_k$)
$H_j$= 1 / (1+ exp(-$\sum_{k=1}^{n}(I_k V_{kj})$))  {compute values of hidden layer units}
$O_i$ = 1 /(1+ exp(- $\sum_{j=1}^{n}(H_j W_j)$))  {compute values of output layer units}
Ranking Vectors ($X_i$) ascending by Output unit ($O_i$)
For each vectors ($X_i$)
E_$O_l$ = $O_l$ * (1 - $O_l$) * ($T_l$- $O_l$)
RMSE = RMSE + root square(($O_l$- $T_l$)$^2$)
MEV = Vector no of (maximum of E_$O_i$)
Set RMSE= RMSE / no of Vectors
if (RMSE equal or less than error_learning) or (max iteration) then
if (RMSE equal or less than error_learning) then
exit and the neural network is learned (set learning-condition to true )
else
Goto step 4 {to start with new initial weights}

#Backward Phase#
else
for each hidden layer unit $H_j$ find error (E_$H_j$)
```

```
$E\_H_{(j)} = H_{(mev)(j)} * (1 - H_{(mev)(j)}) * E\_O_{mev} * W_j$
For each hidden layer unit j
$W_{(j)} = \eta * E\_O_{mev} * H_{(mev)(j)} + W_j$
For each input layer unit $I_k$
For each hidden layer unit $H_j$
$V_{kj} = \eta * E\_H_j * I_k + V_{kj}$
Incremental iteration by one
```



	Standard BP	Modified BP
No. of Initial weights	20	7
Iteration Learning	400000	375718
Error learning	0.00251	0.00099
Learning Time in minute	34:31	9:03

Figure(14): Comparison of The Experimental Results.



No.	Image	Moment1	Moment2	Moment3	Moment4	Moment5	Moment6	Moment7	Output	Target
1		1.709490	2.928680	0.249630	0.112480	0.018840	0.176540	0.001250	0.480050	0.48
2		1.559890	2.888170	1.787720	0.6961	0.747810	0.622540	-0.281080	0.261370	0.26
3		2.247080	6.109720	11.172020	0.125680	0.098950	0.084850	-0.019640	0.041570	0.04
4		1.214990	1.487460	0.553250	0.076470	0.011680	0.037680	0.005950	0.701990	0.70
5		1.321780	2.050460	0.906730	0.084460	-0.014480	-0.028120	0.000530	0.411570	0.41
6		1.126490	1.2690	0.1797	0.023070	-0.000990	-0.006620	0.0003	0.809540	0.81
7		0.724460	0.5253	0.005640	0.000180	0	0.000010	0	0.998480	1
8		0.9260	1.097580	0.078540	0.013220	0.000040	-0.002260	-0.000320	0.849210	0.85
9		1.644270	2.708790	0.2173	0.143760	0.024920	0.213560	-0.002460	0.550530	0.55
10		1.4807	2.305410	2.948130	0.079670	-0.0321	0.026050	0.028470	0.109530	0.11
11		1.361980	1.912390	0.679840	0.242810	0.084420	0.242220	-0.0249	0.591610	0.59
12		1.225480	1.605090	1.037240	0.2258	0.078420	0.101120	-0.069620	0.440960	0.44
13		1.575230	2.489670	0.370110	0.185630	0.046570	0.146650	-0.041830	0.519140	0.52
14		1.083620	1.222670	0.209120	0.072360	0.0069	-0.0003	-0.005760	0.781580	0.78
15		1.973490	4.030170	6.6305	0.885650	0.083540	0.140530	2.179520	0.070220	0.07
16		0.914510	0.842490	0.015880	0.002710	0	0.001140	0	0.961760	0.96
17		2.5463	12.565020	1.071550	0.468860	0.315760	0.872040	-0.072430	0.181440	0.18
18		1.281920	2.037820	1.5304	0.240810	0.033340	0.0464	-0.0629	0.219440	0.22
19		1.688770	3.653070	0.844060	0.532550	0.289760	0.637330	0.108820	0.298960	0.30
20		0.946240	0.916760	0.239010	0.062660	0.007610	0.027840	0.000620	0.889170	0.89
21		1.448980	2.099680	2.863570	0.047930	-0.002170	-0.002260	0.017080	0.149830	0.15
22		0.9916	1.146670	0.275230	0.069560	0.002210	-0.005830	-0.009350	0.7398	0.74
23		1.283860	1.750520	0.320790	0.010280	-0.000380	-0.005080	0.000020	0.671440	0.67
24		1.236260	1.685760	0.628260	0.200520	0.069060	0.159650	-0.027170	0.6295	0.63
25		1.448750	2.553920	0.906870	0.213690	0.062660	0.147790	-0.058980	0.329180	0.33
26		1.338080	2.075560	1.158840	0.3393	0.200060	0.218880	-0.095260	0.3694	0.37
27		0.744110	0.600080	0.198110	0.015210	0.000640	0.003690	-0.000370	0.921870	0.92

Figure(15): Comparison of The Experimental Results.

Conclusion and Future Works

For better throughput, the Assyrian document should be refined first. Therefore, the proposed system needed the Preprocessing stage to make the input text more clearance. Then, the purified text has to be simplified to its composing components. Thus, the proposed approach contained the Segmentation stage in order to isolate the characters that constituting the input Assyrian text. After that, the segmented characters must be represented by a general form that allows these characters to assume different appearances. Hence, the proposed approach included the Feature Extraction stage to generate seven invariant moments which will be used as the Assyrian character badges for the next stage. Finally, the produced character patterns need to be recognized by the neural network of the Classification stage that is learned through the use of the modified back propagation algorithm to generate the corresponding characters in computerized text file.

The experimental results show the tradeoff between computational and learning time on one hand and the accuracy on the other hand. The moment's values are used as accelerator for both the learning process and the recognition process. In future, suggesting using the Discrete Cosine Transform instead of Moments for more powerful learning and recognition accuracy rate of the modified back propagation neural network.

REFERENCES

[1] N . Sebastian, H. Harald, F. Robert, H. Martin, "Classical Syriac". Glottolog. Leipzig: Max Planck Institute for Evolutionary Anthropology, 2013.

[2] M . K. Hu, “Visual pattern recognition by moment invariants”, IRE Trans. Inform. Theory, Vol. IT-8, pp. 179–187, Feb. 1962.

[3] L . Kotoulas and I. Andreadis "Image analysis using moments", Laboratory of Electronics, Section of Electronics and Information Systems Technology, Department of Electrical and Computer Engineering, Democritus University of Thrace, Xanthi 67100, Greece, {lkotoula, iandread@ee.duth.gr}.

[4] C . Holes, "Dialect, Culture, and Society in Eastern Arabia: Glossary", pp. XXIV–XXVI, 2001.

[5] J . R. Smart, "Tradition and Modernity in Arabic Language And Literature", 2013.

[6] B . Klaus, J. F. Healey (trans), "The Aramaic Language: its distribution and subdivisions", Ruprecht, p. 44, 1986.

[7] A l-kenany S.H, "Image files formats and Measuring Image Quality A comparative study and analysis of commonly used image file formats", IJCCCE, .No.1, Vol.5, University of Technology, 2005.

[8] R . Aditya , R. Srivastava, T. Patnaik, B. Kumar, "A Survey of Feature Extraction and Classification Techniques Used In Character Recognition for Indian Scripts", (IJEAT), VOL(2), NO(3), February 2013.

[9] V . C. Hallur, A. A. Malawade, S. G. Itagi, "Survey on Kannada Digits Recognition Using OCR Technique ((IJARCET), VOL(1), NO(10), December 2012.

[10] S . Saha, S. Basu, M. Nasipuri and D. K. Basu,"A Hough Transform based Technique for Text Segmentation", Journal Of Computing, Vol.(2), NO.(2), February 2010.

[11] T . Igarashi, N. Max, and F. Sillion (Guest Editors), "Perceptual Evaluation of Color-to-Gray scale Image Conversions", Czech Technical University in Prague, Czech Republic, Pacific Graphics 2008.

[12] J . Hussain and Lalthlamuana, "Artificial neural network-based approach for Mizo character recognition system", Sci Vis Vol 14 Issue No 2, April-June 2014.

[13] P . Poovizhi, "A Study on Preprocessing Techniques for the Character Recognition", International Journal of Open Information Technologies, Vol. 2, No. 12, 2014.

[14] V . Shrivastava and N. Sharma, "Artificial Neural Network Based Optical Character Recognition", Signal & Image Processing: An International Journal (SIPIJ) Vol.3, No.5, October 2012.

[15] M .L.M Karunanayaka, C.A Marasinghe, N.D Kodikara, "Thresholding, Noise Reduction and Skew correction of Sinhala Handwritten Words", MVA2005 IAPR Conference on Machine VIsion Applications, May 16-18, 2005 Tsukuba Science City, Japan.

[16] D . Gupta, L. Madhu Nair, "Improving Ocr By Effective Pre-Processing And Segmentation For Devanagiri Script:A Quantified Study", 20th June 2013. Vol.52 No.2 2005 - 2013 .

[17] A . M. Raid, W. M. Khedr, M. A. El-dosuky and Mona Aoud, "Image Restoration Based On Morphological Operations", International Journal of Computer Science, Engineering and Information Technology (IJCSEIT), Vol. 4, No.3, June 2014.

[18] H . M. Al-Bayatti, A. S. Rahma, F.A. Al-adhadh, "Document Skew Detection and Correction Algorithm using Wavelet and Radon Transforms", J. of Al-Anbar university for pure science, Vol.3, No.1, 2009.

[19] M . Kaur, S. Jindal, "An Integrated Skew Detection And Correction Using Fast Fourier Transform And DCT", International Journal Of Scientific & Technology Research Volume 2, Issue 12, December 2013.

[20] S . Parashar,S. Sogi, "Finding skewness and deskewing scanned document", Department of Computer Sc. & Engineering, H.C.T.M, Kaithal, Haryana (India), IJCTA, July-August 2012.

[21] A . S. Karnea and S. S. Navalgunda, "Implementation of an Image Thinning Algorithm using Verilog and MATLAB", 2013 INPRESSCO.

[22] K . Saeed, M. Tabe، M. Rybnik, M. Adamski, "A Universal Algorithm For Image Skeletonization And A Review Of Thinning Techniques", Int. J. Appl. Math. Comput. Sci., 2010, Vol. 20, No. 2, 317–335.

[23] K . Kaur M. Sharma, "A Method for Binary Image Thinning using Gradient and Watershed Algorithm", International Journal of Advanced Research in Computer Science and Software Engineering Volume 3, Issue 1, January 2013 ISSN: 2277 128X .

[24] L . Eikvil, "Optical Character Recognition" , December 1993.

[25] M Swamy Das, C. Reddy, A. Govardhan, G. Saikrishna, M. S.Das, "Segmentation Of Overlapping Text Lines, Characters In Printed Telugu Text Document Images", International Journal of Engineering Science and Technology, Vol. 2(11), 2010, 6606-6610.

[26] R . S. Kunte And R. D. Sudhaker Samuel, "A simple and efficient optical character recognition system for basic symbols in printed Kannada text", Sadhana Vol. 32, Part 5, October 2007, pp. 521–533.

[27] O . P. Sharma, M. K. Ghose, K. B. Shah, B. K. Thakur, International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-2, Issue-6, January 2013 Recent Trends and Tools for Feature Extraction in OCR Technology.

[28] M . Zahid Hossain_ M. Ashraful Amin Hong Yan, "Rapid Feature Extraction for Optical Character Recognition", June 4, 2012.

[29] R . Singh, M. Kaur, "OCR for Telugu Script Using Back Propagation Based Classifier", International Journal of Information Technology and Knowledge Management July-December 2010, Volume 2, No. 2, pp. 639-643.

[30] A . Ganatra, Y. P. Kosta, G. Panchal, C. Gajjar, "Initial Classification Through Back Propagation In a Neural Network Following Optimization Through GA to Evaluate the Fitness of an Algorithm", International Journal of Computer Science & Information Technology (IJCSIT), Vol 3, No 1, Feb 2011.

[31] G . Ahmed Mohammed, "Conversion Of English Characters Into Braille Using Neural Network", IJCCCE, NO.2, VOL.11, University of Technology, January 2011.

[32] S . C. Joshi, A. N. Cheeran "MATLAB Based Back-Propagation Neural Network for Automatic Speech Recognition", International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering Vol. 3, Issue 7, July 2014.

[33] C . Zhang, "Character Recognition using a Multilayer Feed forward Neural Network and Back Propagation", Department of Electrical Engineering and Computer Science, University of Tennessee, Knoxville, TN 37996-3450, USA.

[34] M . Saduf, A. Wani, "Comparative Study of Back Propagation Learning Algorithms for Neural Networks", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 12, December 2013.