ABSTRACT

Digits are used in different types of data like vehicle number plate, numeric data, written data, meter reading etc. Digit recognition plays an important role in many user authentication applications in the modern world. In the proposed work, back propagation neural network based digit recognition system has been developed. Such system has four phases: preprocessing, segmentation, feature extraction and classification. When digit image is scanned, quality of image is degraded and some noise is added into this image. So it is necessary to reduce the noise and improve the quality of the digit image for OCR system. For this purpose, image enhancement is necessary. For this, frequency domain based Gaussian filter is used to improve the quality and denoise the digit image. The goal of image segmentation is to separate the clear digit print area from the non-digit area. After segmentation, binary digit image is skeleton to reduce the width of digit into just a single line. In proposed system, coordinates bounding box, area, centroid, eccentricity, equiv-diameter features are used by back propagation neural network as a classifier to recognize digits. Experiments have been performed on standard dataset of digits. Results of this study are quite promising and 96.6% accuracy has been achieved by the proposed system.

Keywords: Back propagation, Gaussian filter, Segmentation, Skeleton.

1. INTRODUCTION

Object character recognition (OCR) systems can contribute tremendously to the advancement of the automation process and can improve the interaction between man and machine in many applications, including office automation, check verification and a large variety of banking, business and data entry applications. The recognition rate of OCR (Optical Character Recognition) systems often decreases for degraded input digit images, such as stroke-connected, stroke-broken and noisy images. This can be caused by a lot of reasons, such as the poor quality of the source paper material, the bad quality of printing, the wrong threshold selected for scanning. The recognition process of an OCR system can be divided into several steps like, preprocessing, segmentation, feature extraction and classification.

History of OCR, has been discussed by Shunji Mori et.al [1]. Toshiyuki Sakai et.al [2] described document information processing both on hardware and software. Faisal Farooq et. al. [3] defined the role of pre-processing in the handwritten recognition. They say that to improve the readability and the automatic recognition of handwritten document images, preprocessing steps are imperative. Hongwei Kong et. al. [4] presented unified method for preprocessing binary text image based on mathematical morphology while S.N. Nawaz et. al [5], Attaullah Khawaja et.al [6] define the basic use of preprocessing in the character recognition. Yungang Zhang and Changshui Zhang [7] define enhancement technique for image preprocessing. They used normalization and enhancement technique for preprocessing. M. Sarfraz et.al [8] proposed a new technique for skew estimation of image document. Pei-Yung Hsiao. et.al [9] in the paper “Generic 2-D Gaussian Smoothing Filter for Noisy Image Processing” define the use of Gaussian filter in OCR. G. Deng and L.W. Cahill [10] propose an adaptive Gaussian filtering algorithm in which the filter variance is adapted to both the noise characteristics and the local variance of the signal. H. Wehbi et. al. [11] provide two segmentation modules, the first one is to isolate the word drawings within a sentence, and the other one is to separate numeral characters and capital letters from a mixed writing prior to their recognition. Xiaoping Li et. al. [12] also proposed the connected domain analysis method, segmentation projection histogram method and shelling projection method for segmentation of digits. M. Blumenstein et. al. [13] introduced new feature extraction technique investigated whose research was based on the calculation and location of transition features from background to foreground pixels in the vertical and horizontal directions. Brijesh Verma et.al [14] proposed feature extraction technique that can be classified into eight modules such as dehooking, extract feature points, stroke extracting, calculate PEN-UP, extract zones and directions of start point and end point, extract changes in writing direction,
calculate height width ratio and extract zone information which creates a global feature vector and uses a back-propagation neural network based classifier. S.V. Rajashekararadhya and Vanaja Ranjan P [15] propose a zone-based hybrid feature extraction system. R.J. Ramteke et.al [16] present an experimental evaluation of the effectiveness of various techniques based upon moment invariants. Douglas Kozlay [17] describes a feature extraction technique for the next generation of optical character readers. Hermineh Sanossian [18] tells us that extract relevant features from numeral images reduce the complexity of recognition. He introduces the new method in his paper. He finds different features, features due to boundary distances in a segment the pixel densities in a segment, and line distances from centroid in a segment. Chongliang Zhong et.al [19] introduce the 13-point feature of skeleton method. There are two steps that they take. One is to find the skeleton of the character and the other is to extract the 13-point feature of skeleton. S.V. Rajashekararadhya et.al [20] present Zone and Distance metric based feature extraction system. Abdur Rahim Md. Forkan et.al. [21] describe in the paper that how multi layered feed forward artificial neural network for classification of character and position of white pixels of the images considered as noise and high pass filtering is used to remove this noise from input image. Zaheer Ahmad et.al [22] developed a system consists of two main modules segmentation and classification. Ping Zhang, Lihui Chen, Alex C Kot [23] proposed a hybrid neural network and tree classification system for handwritten numeral recognition. Christopher L. Scofield, Lannie Kenton, Jung-Chou Chang [24] proposed a multiple neural network system (MNNS) for image-based character recognition. Based on artificial neural network, digital image processing, and features extraction theory, Huang Hanmin Huang et.al [25] analyzed BP network’s affect and presented its improving solutions. Hermineh Y.Y. Sanossian [26] described an optical character recognition system which uses multilayer perceptron classifier. Michael D. Garris [27] defines fourier descriptors, moment invariants, and other boundary features. Adrian Lim Hooi Jin et.al [28] say that all the processed fingerprint image are used as an input to the back propagation neural network designed to perform the training process.

Thus, a large number of researchers have published paper on this area each has own advantage and disadvantage. In fact the need of hour is to develop a digit recognition system for improving recognition accuracy. So in the present work back propagation neural network based digit recognition system has been proposed.

2. PROPOSED WORK

Proposed digit recognition system has four phases:

2.1 Pre processing
2.2 Segmentation
2.3 Feature extraction
2.4 Classification

This system uses standard database that is free available on internet. Some sample of database:

Fig. 1: Steps for Digit Recognition System

Fig. 2: A Set of Typical Digit Images from Database
2.1 Pre-Processing
When digit image is scanned, quality of image is degraded and some noise is added into the image. These parameters can reduce the accuracy of OCR system. So it is necessary to reduce the noise and improve the quality of the digit image for OCR system. So main objective of preprocessing phase is to denoise and enhance the quality of scanned digit image. For this purpose, the following proposed approach is used:

Suppose that given image is \( f(x, y) \) of size MxN.

1. Multiply the input image by \((-1)^{x+y}\) to center the transform.

\[
f(x, y) \rightarrow (-1)^{x+y}
\]

2. Compute Discrete Fourier transform (DFT) of \( f(x, y) \)

\[
F[u, v] = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi \left( \frac{x}{M} + \frac{y}{N} \right)}
\]

3. Compute DFT of the given kernel which is denoted by \( H(u, v) \).

4. Multiply the two transforms together. Multiplication is done on a point by point basis which means that the transforms of the kernel and the image must have the same dimensions. This is done by padding out the kernel with zero prior to computing its transform.

\[
G(u, v) = F\left( \frac{u}{M}, \frac{v}{N} \right) H(u, v)
\]

5. Compute the Inverse DFT of product \( G(u, v) \).

\[
g'(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} G(u, v) e^{j2\pi \left( \frac{ux}{M} + \frac{vy}{N} \right)}
\]

6. Multiply \( g'(x, y) \) by \((-1)^{x+y}\) to get enhanced and denoised image \( g(x, y) \).

In the proposed approach gaussian filter is used to enhance the image and to remove the noise from the image.

\[
H(u, v) = e^{-D^2(u, v) / 2\sigma^2}
\]

Where \( D(u, v) \) is the distance from the origin of the Fourier transform, \( \sigma \) is a measure of the spread of the Gaussian curve.

2.2 Segmentation
Image segmentation is one of the most important steps leading to the analysis of processed image data. Its main goal is to divide an image into parts that have a strong correlation with objects or areas of the real world contained in the image. Segmentation is crucial for recognition system. Objective of segmentation is to partition an image into regions. The main objective of segmentation of digit image is to separate the clear digit print area from the non-digit area. For this purpose, suitable threshold is needed [29]. Threshold creates input digit image into a binarized digit image.

A thresholded digit image \( g(x, y) \) is defined as

\[
g(x, y) = \begin{cases} 1 & \text{if } f(x, y) > T \\ 0 & \text{if } f(x, y) \geq T \end{cases}
\]

In proposed approach the following algorithm is used to obtain threshold value.

1. Let \( T \) be initial Threshold value for given image.
2. Segment the image into two regions \( R_1 \) and \( R_2 \) using \( T \). Where Region \( R_1 \) is a group of pixels whose gray level values > \( T \) and \( R_2 \) consisting of pixels with values \( \leq T \).
3. Compute the average gray level values \( \mu_1 \) and \( \mu_2 \) for the pixels in regions \( R_1 \) and \( R_2 \).
4. Compute a new threshold value

\[
T = \left( \frac{\mu_1 + \mu_2}{2} \right)
\]

5. Repeat step 2 through 4 until the difference in \( T \) in successive iterations is smaller than a predefined parameter \( T_0 \).

Image segmentation is to separate the clear digit print area from the non-digit area. After segmentation, binary digit image is skeleton to reduce the width of digit into just a single line. The flowchart for all steps up to skeletonization in OCR System is clearly illustrated in Figure 3.

---

**Flowchart of the Four Image Processing Steps and Its Respective Results**

- **Start**
- **Grab digit image.**
- **Image Enhancement**
- **Segmentation**
- **Skeletonization.**
- **Stop**

---

[Figure 3: Flowchart of the Four Image Processing Steps and Its Respective Results]
2.3 Feature Extraction
To find all the features, first take the smallest rectangle that cover the digit region. After this, the following features are extracted:

2.3.1 Bounding Box
Bounding box feature provide 4 outputs. One is the top left x-coordinate of the smallest rectangle that cover the digit. Second is the top left y-coordinate of the smallest rectangle that cover the digit. Third is lower right x-coordinate of the smallest rectangle that covers the digit. Fourth is downright y-coordinate of the smallest rectangle that covers the digit.

2.3.2 Area
To find the area, count the number of pixels in that region.

\[ A = \text{Number of pixels in the smallest rectangle that cover the digit.} \]

2.3.3 Centroid
Centroid is the center of mass of the region. It provides two outputs first is the horizontal coordinate of the center of mass and second is the vertical coordinate of the center of mass.

\[
\bar{m} = \frac{1}{N} \sum_{(m, n) \in R} m
\]
\[
\bar{n} = \frac{1}{M} \sum_{(m, n) \in R} n
\]

To find the remaining features, take best fit ellipse that lie inside the smallest rectangle which cover the digit region. The best fit ellipse is the ellipse whose second moment equals to that of the digit.

The \((p, q)\) order central moments are

\[
\mu_{p,q} = \sum_{(m, n) \in R} (m - \bar{m})^p (n - \bar{n})^q
\]

2.3.4 Major Axis Length
Major axis length is the length of major axis of the best fit ellipse.

2.3.5 Minor Axis Length
Minor axis length is the length of minor axis of the best fit ellipse.

2.3.6 Eccentricity
Eccentricity [30] is defined as

\[
\varepsilon = \frac{(\mu_{2,0} - \mu_{1,0})^2 + 4\mu_{1,1}}{\text{Area}}
\]

The eccentricity is the ratio of the distance between the foci of the ellipse and its major axis length. The value is between 0 and 1. An ellipse whose eccentricity is 0 is actually a circle, while an ellipse whose eccentricity is 1 is a line segment.

To find the Equiv-diameter features, take best fit circle that lie inside the smallest rectangle which cover the digit region.

2.3.7 Equiv-diameter
The diameter of a circle with the same area as the region. Equiv-diameter computed as

\[
\text{Equiv-diameter} = \sqrt{(4 \times \text{Area})/\pi}
\]

2.4 Classification
The classification stage is the main decision making stage of a digits recognition system. This stage uses the features extracted in the previous stage to identify the digits. In this stage, artificial neural network is normally used.

In the proposed work, particularly back propagation neural network model [31] of artificial neural networks is proposed for classification of digits.

2.4.1 Architecture of Back Propagation for Digit Recognition
In the proposed system, back propagation neural network has one input layer, one hidden layer and one output layer. The number of units (neurons) in the input layer is determined by the length of the training input vectors. In the proposed system, ten features are used in the training vector. So, the number of units in the input layer is ten which are denoted by \(X_1 \ldots X_{10}\). Hidden layer has fifteen neurons which are denoted by \(Z_1 \ldots Z_{15}\). Output layer has one unit which is denoted by \(Y_1\). The bias on a typical output unit \(Y_j\) is denoted by \(V_{0j}\) and the bias on a typical hidden unit \(Z_j\) (\(j = 1, \ldots, 15\)) is denoted \(V_{0j}\) (where \(j = 1\ldots15\)). Weights between the hidden layer and the input layer are denoted by \(W_{ij}\) (where \(i = 1\ldots10\) and \(j = 1\ldots15\)) and weights between the hidden layer and the output layer are denoted by \(W_{1j}\) (where \(j = 1\ldots15\)).

During feed forward, each input unit receives an input signal and sends this signal to the each of the hidden units. Each hidden unit then computes its activation and sends its signal to each output unit. Output unit computes its activation to form the response of the net for the given input pattern.

During training, each output unit compares its computed activation \(y_i\) with its target value \(t_i\) to determine the associated error for that pattern with that unit. Based on this error, the factor \(d_i\) is computed. \(d_i\) is used to distribute the error at output unit \(Y_j\) back to all units in the previous layer. It is also used (later) to update the weights between the output and the hidden layer. In a similar manner, the factor \(d_j\) (\(j = 1\ldots15\)) is computed for each hidden unit \(Z_j\). It is not necessary to propagate the error back to the input layer, but \(d_j\) is used to update the weights between the hidden layer and the input layer.
After this all of the $d$ factors have been determined, the weights corresponding to all layers are changed simultaneously. The adjustment to the weight $W_{jl}$ (from hidden unit $Z_j$ to output unit $Y_1$) is based on the factor $d_1$ and the activation $z_j$ of the hidden unit $Z_j$. The changes to the weight $V_{ij}$ (from input unit $X_i$ to hidden unit $Z_j$) is based on the factor $d_j$ and the activation $x_i$ of the input unit.

The nomenclature we use in the training algorithm for the back propagation net is as follows:

- $X$: Input training vector: $x = (x_1, \ldots, x_{10})$.
- $T$: Output target vector: $t = (t_1)$.
- $d_1$: Portion of error correction weight adjustment for $W_{lj}$ that is due to an error at output unit $Y_1$ also, the information about the error at unit $Y_1$ that is propagated back to the hidden units that feed into unit $Y_1$.
- $d_j$: Portion of error correction weight adjustment for $V_{ij}$ that is due to the back propagation of error information from the output layer to the hidden unit $Z_j$.
- $A$: Learning rate
- $x_i$: Input unit
- $V_{ij}$: Bias on hidden unit $j = 1, \ldots, 15$
- $Z_j$: Hidden unit
- $z_{inj}$: The net input to $Z_j$ is denoted $z_{inj}$
- $y_{in1}$: The output signal (activation) of $Y_1$ is denoted $y_{in1}$

2.4.2 Proposed Algorithm to Train Back Propagation Neural Network for Digit

Step 1. Initialize weights (set to small random values)
Step 2. While stopping condition is false, do step 3-10.

**Feed forward:**
Step 3. For each training pair, do step 4-9.

Step 4. Each input unit ($X_i, i = 1, \ldots, 10$) receives input signal $x_i$ and broadcasts this signal to all units in the layer above (the hidden units).
Step 5. Each hidden unit ($Z_j, j = 1, \ldots, 15$) sums its weighted input signals.

$$z_{inj} = V_{0j} + \sum_{i=1}^{10} x_i V_{ij}$$

Applies its activation function to compute its output signal

$$z_j = f(z_{inj})$$

and sends this signal to all units in the layer above (output units).
Step 6. Each output unit ($Y_1$) sums its weighted input signals,

$$y_{in1} = w_{01} + \sum_{j=1}^{15} z_j W_{j1}$$

and applies its activation function to compute its output signal,

$$y_1 = f(y_{in1})$$

**Back propagation of error:**
Step 7. Output unit ($Y_1$) receives a target pattern corresponding to the input training pattern, Computes its error information term,

$$d_1 = (t_1 - y_1)f'(y_{in1})$$

Calculates its weight correction term (used to update $W_{jk}$ later).

$$\Delta W_{j1} = Ad_1 z_j$$

Calculates its bias correction term (used to update $W_{ok}$ later).

$$\Delta W_{o1} = Ad_1$$

and sends $d_i$ to units in the layer below.
Step 8. Each hidden unit ($Z_j, j = 1, \ldots, 15$) sums its delta inputs (from units in the layer above),
\[
d_{in_j} = d_iW_{ij}
\]

multiplies by the derivative of its activation function to calculate its error information term.

\[
d_j = d_{in_j} f'(z_{in_j})
\]
calculates its weight correction term (used to update \(V_{ij}\) later),

\[
\Delta V_{ij} = Ad_j x_i
\]

and calculates its bias correction term (used to update \(V_{ij}\) later),

\[
\Delta V_{ij} = Ad_j
\]

Update weights and biases:

Step 9. Output unit \((Y_j)\) updates its bias and weights \((j = 0 \ldots 15)\)

\[
W_{ji}^{(new)} = W_{ji}^{(old)} + \Delta W_{ji}
\]

Each hidden unit \((Z_{j}, j = 1\ldots 15)\) updates its bias and weights \((i = 0 \ldots 10)\)

\[
V_{ij}^{(new)} = V_{ij}^{(old)} + \Delta V_{ij}.
\]

Step 10. Test stopping condition.

### 2.4.3 Activation Function

In the proposed study, performance of back propagation algorithm for digits recognition is measured at two different activation functions:

**Sigmoid Function:**

For sigmoid function, the output varies continuously but not linearly as the input changes. This function is a continuous function that varies gradually between asymptotic values 0 and 1 or -1 and +1 and is given by

\[
f(x) = \frac{1}{1 + e^{-ax}}
\]

Where \(a\) is slope parameter, which adjusts the abruptness of the function as it changes between the two asymptotic values. Sigmoid function is differentiable, which is an important feature of neural network theory.

**Hyperbolic Tangent Function**

This activation function is defined as

\[
f(x) = \tanh(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}}
\]

### 2.4.5 Momentum Factor for Digit Recognition

In Backpropagation network (BPN), the weight change is in a direction that is a combination of current gradient and the previous gradient. This approach is beneficial when some training data are very difficult from a majority of data. A small learning rate is used to avoid major description of the direction of learning when very unusual pair of training patterns is presented. If the momentum is added to the weight update formula, the convergence is faster. It is found that momentum allows the net to perform large weight adjustments as long as the correction proceeds in the same general direction for several patterns. Thus using momentum, the net does not proceed in the direction of the gradient, but travels in the direction of the combination of the current gradient and the previous direction for which the weight correction is made. The main purpose of momentum is to accelerate the convergence of error propagation algorithm. This method makes the current weight adjustment with a fraction of the recent weight adjustment

\[
W_{ji}(t + 1) = W_{ji}(t) + ad_j z_j + mf[W_{ji}(t) - W_{ji}(t - 1)]
\]

\[
V_{ij}(t + 1) = V_{ij}(t) + ad_j x_i + mf[V_{ij}(t) - V_{ij}(t - 1)]
\]

\(mf\) is called the momentum factor and it ranges from 0 < \(mf\) < 1.

### 3. Result and Discussion

The digits recognition system using neural network has been implemented using MATLAB version 7.0. In the proposed study, 10 digits from 0 to 9 in the form of various digital images from standard data set are used. First of all images are enhanced by using enhancement approach. After enhancing the images, most suitable features such as Area, Eccentricity, Equidiameter, Minor axis length, Centroid, Bounding are extracted. These features are used to train and test the proposed system.

Values of features are normalized using the following equation.

\[
X_i = \frac{x(i)}{\max \{x(i)\}}
\]

In proposed study, proposed system is first trained with five neurons in the hidden layer by varying the momentum factor and learning rate using sigmoid as activation function. After that total squared errors (TSE) and number of epochs (NE) are found results are shown in Fig 1. It is observed that when five neurons are in hidden layer, more epochs are required for training. After this eight neurons are used in hidden layer to train neural network at different learning rate, momentum factor, results are shown in Fig. Then the same experiments repeated with twelve neurons in hidden layer at various learning rates, momentum factors shown in Fig 3. But when fifteen neurons in hidden layer are used at different learning rates and momentum factors, it is observed that the results are satisfactory. TSE and number of epochs are minimized in this case. With learning rates 0.6 and momentum factor 0.8, system shows TSE = 0.009 with 3500 epochs for training as shown in Fig 4. So it is observed that proposed system give good results when it is trained at learning rate 0.6, momentum factor 0.8 and 15 neurons in the hidden layer for digits recognition.
In proposed study, proposed system is also trained with fifteen neurons in the hidden layer by varying the momentum factor and learning rate using Hyperbolic Tangent function as activation function. But results are not satisfactory.

After exploring various parameters like number of neurons in hidden layer, learning rate and momentum factor to train a neural network, back propagation learning with learning rate 0.6 and momentum factor 0.8 is used to train the network in tolerable time for digits recognition. In this system, the most suitable number of neurons in hidden layer are fifteen and activation function is sigmoid function. After training, performance of proposed system is measured using testing data set whose some sample are shown in table 4.2. The mean absolute percentage error (MAPE) is a parameter used to measure the performance of artificial neural network based system. So, performance of proposed system for Digits recognition is measured by using MAPE. MAPE is defined as:

\[
\text{MAPE} (%) = \frac{1}{N} \sum_{i=1}^{N} \left| \frac{T_D - A_D}{A_D} \right| \times 100
\]

Where \(T_D\) and \(A_D\) are the target digit and actual digit respectively and \(N\) is the number of digits. MAPE for the proposed system is 0.45\%. Actual and Target digits were shown by plotting actual and target digits curves on the same graph as shown in Fig 9.
3.1 Accuracy

Accuracy of the proposed system is calculated by dividing correctly recognized digits by total number of digits which are actually present. In the proposed system, 60 digits are selected from the testing data set for testing. Among which 58 digits are recognized correctly by system. Thus, overall accuracy of the system is 96.6%.

\[
\text{Accuracy} = \frac{\text{Correct recognized number of digits}}{\text{Total number of digits selected for testing}} \times 100
\]

Table 2

<table>
<thead>
<tr>
<th>Target Digits</th>
<th>Output of proposed system</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.0516</td>
</tr>
<tr>
<td>1</td>
<td>1.0673</td>
</tr>
<tr>
<td>2</td>
<td>2.1339</td>
</tr>
<tr>
<td>3</td>
<td>2.7528</td>
</tr>
<tr>
<td>4</td>
<td>4.1251</td>
</tr>
<tr>
<td>5</td>
<td>5.0172</td>
</tr>
<tr>
<td>6</td>
<td>5.8069</td>
</tr>
<tr>
<td>7</td>
<td>6.7238</td>
</tr>
<tr>
<td>8</td>
<td>7.8632</td>
</tr>
<tr>
<td>9</td>
<td>8.6112</td>
</tr>
</tbody>
</table>

4. CONCLUSION

Back propagation results shown above are satisfactory for digits recognition when we used sigmoid activation function instead of hyperbolic tangent. Momentum factor = 0.8 and learning rate = 0.6 are best for this. The present study verifies that the neural network prediction is nearly same as the actual with mean approximate percentage error of 1.12 for Digits Recognition. In the proposed system, 60 digits are selected from the testing data set for testing. Among which 58 digits are recognized correctly by system. Thus, overall accuracy of the system is 96.6%. Our future work will include the recognition of text and other images.

REFERENCES


