

PAPER • OPEN ACCESS

## Handwritten Arabic numerals recognition system using probabilistic neural networks

To cite this article: Yushun Tang *et al* 2021 *J. Phys.: Conf. Ser.* **1738** 012082

View the [article online](#) for updates and enhancements.



The Electrochemical Society  
Advancing solid state & electrochemical science & technology

**240th ECS Meeting** ORLANDO, FL

Orange County Convention Center Oct 10-14, 2021



Abstract submission due: April 9

**SUBMIT NOW**

# Handwritten Arabic numerals recognition system using probabilistic neural networks

Yushun Tang<sup>1</sup>, Shengguo Zhang<sup>1,2\*</sup> and Lu Niu<sup>1</sup>

<sup>1</sup> Key Laboratory of China's Ethnic Languages and Intelligent Processing of Gansu Province, Northwest Minzu University, Lanzhou, Gansu 730000, China

<sup>2</sup> Department of Electrical Engineering, Northwest University for Nationalities, Northwest Minzu University, Lanzhou, Gansu, 730030, China

\*Corresponding author's e-mail: zhangshengguo@tsinghua.org.cn

**Abstract.** This paper presents a system for the recognition of the handwritten Arabic numerals zero to nine (0–9) using a probabilistic neural network (PNN) approach. This system can recognize handwritten input and externally imported Arabic numerals in real time, including two processes of image pre-processing and recognition. Image pre-processing involved normalization and expansion to enlarge the characteristics of the picture for easy recognition. Recognition process involves the calculation of mode distance, which can help obtain the similarity match between sample matrix and learning matrix. According to similarity matching the input feature matrix will be classified into one of the ten numbers. This study uses MATLAB to establish a graphical interface that is easy to operate. The system has good performance and strong scalability, which established a simple experimental platform for a more in-depth study of the recognition of handwritten Arabic numerals.

## 1. Introduction

The recognition technology of handwritten Arabic numerals has been a research hotspot for many years. It is an important field of image processing and character recognition. Due to the widespread use of Arabic numerals, this technology is very widely applied in specific environments, such as automatic identification of express number or postal code, automatic processing systems for bank checks, which are the core and key to some automated systems. The study of handwritten Arabic numerals recognition is of great significance and wide application.

With the increasingly fierce research on artificial neural networks, artificial neural networks have brought new recognition ways to handwritten Arabic numeral recognition due to their characteristics of strong anti-interference, strong robustness, high fault tolerance, self-adaptation and strong self-learning ability etc., The recognition system using artificial neural network has the characteristics of fast recognition speed and low recognition error rate, so it has a large number of applications in the field of character recognition.

## 2. Probabilistic neural network

Probabilistic neural network for short PNN, as shown in Figure (1) below, which consists of four layers: input layer, pattern layer, summation layer and output layer.



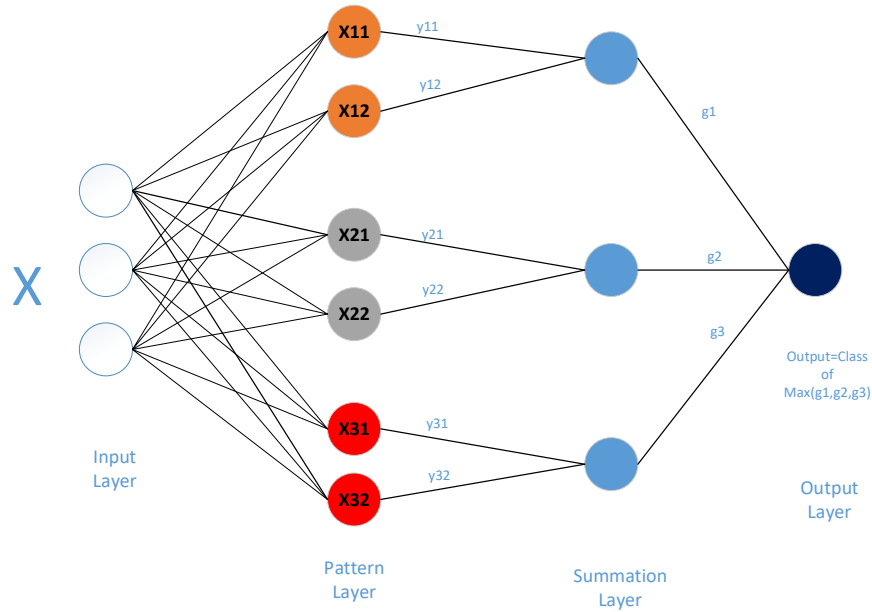


Figure (1). Probabilistic neural network structure

Input layer, used to receive input matrix from pre-processed picture and pass feature vectors to the network. Pattern layer receives the data transferred from the input layer and calculates the distance between the feature matrix and training sets. This distance represents the initial probability that the feature matrix matches each pattern in the training set. Summation layer accumulates the probabilities of each class, and then calculate the best matching type from the probability transfer function. The final result will be output in the output layer, and only one of the N summing layer neuron output results is 1, which is the type with the largest probability value, and the remaining results are all 0.

### 2.1. Bayesian decision principle

Probabilistic neural network is a feedforward artificial neural network based on the Bayesian decision and statistical principle, whose working mode mainly depends on Bayesian classifier. Bayesian classifier theory is If  $p(w_i | \vec{x}) > p(w_j | \vec{x}) \quad \forall j \neq i$ , then  $\vec{x} \in w_i$  and  $p(w_i | \vec{x}) = p(w_i)p(\vec{x} | w_i)$ . Generally speaking, the probability density function  $[p(w_i | \vec{x})]$  of each class is unknown but can be calculated by Bayesian decision theory using Parzen function. As shown in the following formula(1):

$$p(\vec{x} | w_i) = \frac{1}{N_i} \sum_{k=1}^{N_i} \frac{1}{2\pi^{\frac{l}{2}} \sigma^l} \exp\left(-\frac{\|\vec{x} - \vec{x}_{ik}\|^2}{2\sigma^2}\right) \quad (1)$$

$\vec{x}_{ik}$  is the k-th training sample of the i-th class,  $l$  is the dimension of the sample,  $\sigma$  is the smoothing parameter,  $N_i$  is the total number of i-th training samples.

### 2.2. Calculation of mode distance

The mode distance refers to the disparity between corresponding elements of sample matrix and learning matrix, size represents the closeness of two matrices, which is an important part of probability matching. Calculate the mode distance in pattern layer. Before this process, pattern layer will receive some normalized data from input layer. They are called learning matrix. Assuming  $m$  inputs, each input has  $n$  characteristic attributes, normalized data is shown in the following formula(2):

$$L_{m*n} = \begin{pmatrix} \frac{x_{11}}{\sqrt{M_1}} & \frac{x_{12}}{\sqrt{M_1}} & \cdots & \frac{x_{1n}}{\sqrt{M_1}} \\ \frac{x_{21}}{\sqrt{M_2}} & \frac{x_{22}}{\sqrt{M_2}} & \cdots & \frac{x_{2n}}{\sqrt{M_2}} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{x_{m1}}{\sqrt{M_m}} & \frac{x_{m2}}{\sqrt{M_m}} & \cdots & \frac{x_{mn}}{\sqrt{M_m}} \end{pmatrix} = \begin{pmatrix} l_{11} & l_{12} & \cdots & l_{1n} \\ l_{21} & l_{22} & \cdots & l_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ l_{m1} & l_{m2} & \cdots & l_{mn} \end{pmatrix} \quad (2)$$

$x_{mn}$  is the input matrix,  $M_1 = \sum_{k=1}^n x_{1k}^2, M_2 = \sum_{k=1}^n x_{2k}^2, \dots, M_m = \sum_{k=1}^n x_{mk}^2$ . The normalized data can make the mode distance calculation more convenient and reduce errors. The sample matrix is

$$S_{m*n} = \begin{pmatrix} s_{11} & s_{12} & \cdots & s_{1n} \\ s_{21} & s_{22} & \cdots & s_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ s_{m1} & s_{m2} & \cdots & s_{mn} \end{pmatrix}$$

normalized and expressed in the same way. Then using the Euclidean distance theory to calculate mode distance between learning matrix and sample matrix as following formula (3):

$$D_{m*n} = \begin{pmatrix} \sqrt{\sum_{k=1}^n |s_{1k} - l_{1k}|^2} & \sqrt{\sum_{k=1}^n |s_{1k} - l_{2k}|^2} & \cdots & \sqrt{\sum_{k=1}^n |s_{1k} - l_{mk}|^2} \\ \sqrt{\sum_{k=1}^n |s_{2k} - l_{1k}|^2} & \sqrt{\sum_{k=1}^n |s_{2k} - l_{2k}|^2} & \cdots & \sqrt{\sum_{k=1}^n |s_{2k} - l_{mk}|^2} \\ \cdots & \cdots & \cdots & \cdots \\ \sqrt{\sum_{k=1}^n |s_{mk} - l_{1k}|^2} & \sqrt{\sum_{k=1}^n |s_{mk} - l_{2k}|^2} & \cdots & \sqrt{\sum_{k=1}^n |s_{mk} - l_{mk}|^2} \end{pmatrix} = \begin{pmatrix} d_{11} & d_{12} & \cdots & d_{1n} \\ d_{21} & d_{22} & \cdots & d_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ d_{m1} & d_{m2} & \cdots & d_{mn} \end{pmatrix} \quad (3)$$

### 3. Probability calculation

This process is on the summation layer. Neurons in the summation layer will be activated after receiving the mode distance. Take the Gaussian function with standard deviation  $\sigma=0.1$  and activate to get the initial probability matrix as following formula(4):

$$Pi_{m*n} = \begin{pmatrix} e^{-\frac{d_{11}}{2\sigma^2}} & e^{-\frac{d_{12}}{2\sigma^2}} & \cdots & e^{-\frac{d_{1n}}{2\sigma^2}} \\ e^{-\frac{d_{21}}{2\sigma^2}} & e^{-\frac{d_{22}}{2\sigma^2}} & \cdots & e^{-\frac{d_{2n}}{2\sigma^2}} \\ \cdots & \cdots & \cdots & \cdots \\ e^{-\frac{d_{m1}}{2\sigma^2}} & e^{-\frac{d_{m2}}{2\sigma^2}} & \cdots & e^{-\frac{d_{mn}}{2\sigma^2}} \end{pmatrix} = \begin{pmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ p_{m1} & p_{m2} & \cdots & p_{mn} \end{pmatrix} \quad (4)$$

Assuming a samples, which can be divided into c class, and the number of samples of each type is the same, set to k, then find the sum of the initial probabilities that each sample belongs to various types as following formula(5):

$$Pt_{m*n} = \begin{pmatrix} \sum_{b=1}^k p_{1b} & \sum_{b=k+1}^{2k} p_{1b} & \cdots & \sum_{b=a-k+1}^a p_{1b} \\ \sum_{b=1}^k p_{2b} & \sum_{b=k+1}^{2k} p_{2b} & \cdots & \sum_{b=a-k+1}^a p_{2b} \\ \cdots & \cdots & \cdots & \cdots \\ \sum_{b=1}^k p_{mb} & \sum_{b=k+1}^{2k} p_{mb} & \cdots & \sum_{b=a-k+1}^a p_{mb} \end{pmatrix} = \begin{pmatrix} t_{11} & t_{12} & \cdots & t_{1c} \\ t_{21} & t_{22} & \cdots & t_{2c} \\ \cdots & \cdots & \cdots & \cdots \\ t_{m1} & t_{m2} & \cdots & t_{mc} \end{pmatrix} \quad (5)$$

Finally, the probability following formula (6):

$$prob_{ij} = \frac{t_{ij}}{\sum_{k=1}^c t_{ik}} \quad (6)$$

$t_{ij}$  is sum of the initial probabilities that i-th sample belonging to the j-th class.

#### 4. Implementation of handwritten Arabic numerals recognition system

This paper uses the GUI toolbox in MATLAB to build an easy-to-use graphical interface to implement system functions. The system mainly includes 2 modules of handwriting panel and recognition panel.

##### 4.1. Handwriting panel

There are two ways to collect samples, one is real-time handwriting input through the handwriting panel, and the other is to import pictures externally. This panel can collect handwritten data as following figure (2 a, b):

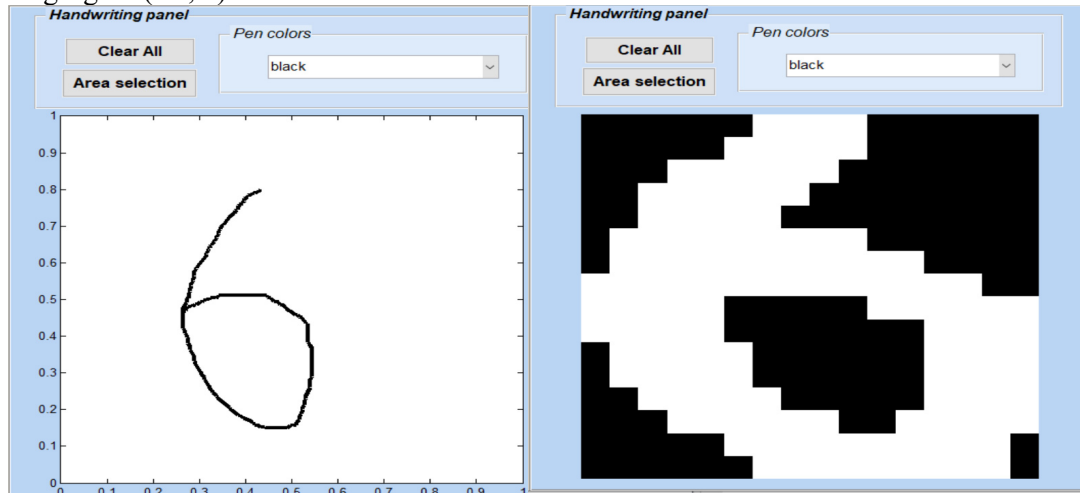


Figure (2 a).  
Handwritten data input

Figure (2 b).  
Pre-processed data input

Figure (2 a) shows input Arabic numeral 6, and Figure (2 b) shows the result of pre-process. Pre-processing by dilating the binarized image to make the image into a 16\*16 pixel picture. They will be sent to the input layer of the PNN in the form of a matrix, where the white pixels are 0 and the black pixels are 1. Handwriting input is available in 7 colors, figure (2 a) selects the black. When multiple numbers are written on the panel at the same time, area selection can recognize them one by one.

##### 4.2. Recognition panel

After image pre-processing, the input is a 256-dimensional feature matrix, and the number of neuron nodes in input layer and pattern layer of PNN is 256. Since the target to be recognized is the 10 Arabic

numerals 0-9. The number of neurons in the summation layer is 10, Finally, the result calculated by the probability value depend on the class with the highest probability value, the output is 1 and the rest is 0. Only the result is 1 will be output. As shown in the figure (3) below, the first column is the handwritten input number, the second is the number pre-processing result, and the third one is the recognition result.

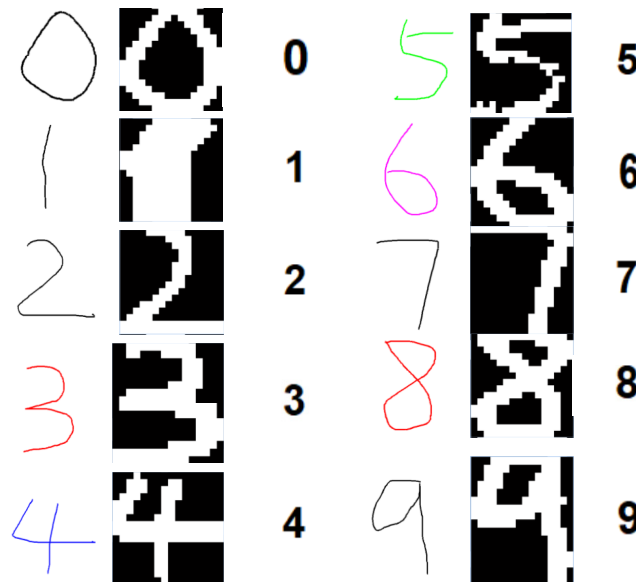


Figure (3). Recognition result structure

## 5. Conclusion

This paper uses MATLAB to implement a simple handwritten Arabic numeral recognition system, using PNN as the system classifier. The main recognition process includes binarization of data input and dilation pre-processing, mode distance and probability value calculation. At the same time, the system uses GUI toolbox in MATLAB to establish a convenient graphical operation interface, which can effectively recognize handwritten Arabic numerals in real time. However, there are several shortages: First, writing numbers or externally input pictures cannot be too scribbled and tilted, too scribbled cannot be recognized, there is no tilt correction in the system recognition algorithm, so cannot recognize the tilt one. The second is that pictures imported externally can only be picture data of specific pixels (16\*16). For the first problem, tilt correction and other various feature extraction methods can be added to the recognition algorithm to enhance the system recognition rate. For the second problem, a pre-processing process can be added before the data collection process to convert externally imported pictures into 16\*16 pixel pictures. This system has good scalability and establishes a simple experimental platform for a more in-depth study of the recognition of handwritten Arabic numerals.

## Acknowledgments

This work was supported in part by National Natural Science Foundation of China (Grant No. 51465053) and in part by Fundamental Research Funds for the Central Universities of China (Grant No. 31920200043).

## References

- [1] Ling, T., Lei, L. (2020) Automatic recognition of handwritten digits in metrology based on matlab. *Industrial Metrology*, 1: 43-45.
- [2] Changtong, S., Liming, H., Hui, W. (2016) Handwritten digit recognition based on probabilistic neural network. *Microcomputer applications*, 10: 14-15.

- [3] Jianju, X., Jun, L., Zhanfu, X. (2016) Application of PNN in Handwritten Digit Recognition. Modern Computer: Mid-term, 8:20-23.
- [4] Huiying, L., Xichuan, H. (2018) Feature extraction of handwritten numerals based on probabilistic neural network. Modern computer: early and late, 10: 59-63
- [5] Yuanyuan, C., Huanli, Y., Qishuang, S. (2016) Handwritten digit recognition based on neural network. Intelligent computer and applications, 3: 140-141