

HANDWRITING RECOGNITION ON IMAGE

Tomáš Kováčik¹, Daniela Ďuračková¹, Vladimír Sedlák¹, Roman Zálusky¹

*¹ Institute of Electronics and Photonics, Faculty of Electrical Engineering and
Information Technology, Ilkovicova 3,
812 19 Bratislava, Slovakia
E-mail: tomas_kovacik@stuba.sk*

Received 30 April 2015; accepted 16 May 2015

1. Abstract

The Area of Signal processing and image recognition is nowadays an object of research not only in academic field but also in commercial area. This part of research is and will be a part of our life. This paper focuses on creating algorithm, which is able learn and recognize handwritten numbers and in second step can identify these numbers in document (in our case document is .jpg image Fig.5). Because each people have specific pattern of handwritten numbers in our work we focuses on author handwritten. At first we created database which consist from 3690 patterns (369 patterns to one number) and 260 characters from alphabet (8 characters to one letter). And we use these databases in our work.

2. Neural Network

The human brain can be described as a biological neural network—an interconnected web of neurons transmitting elaborate patterns of electrical signals. Dendrites receive input signals and, based on those inputs, fire an output signal via an axon. Computer scientists have long been inspired by the human brain. In 1943, Warren S. McCulloch, a neuroscientist, and Walter Pitts, a logician, developed the first conceptual model of an artificial neural network. They describe the concept of a neuron, a single cell living in a network of cells that receives inputs, processes those inputs, and generates an output. Their work, and the work of many scientists and researchers that followed, was not meant to accurately describe how the biological brain works. Rather, an artificial neural network (which we will now simply refer to as a “neural network”) was designed as a computational model based on the brain to solve certain kinds of problems. A neural network is a “connectionist” computational system. The computational systems we write are procedural; a program starts at the first line of code, executes it, and goes on to the next, following instructions in a linear fashion. A true neural network does not follow a linear path. Rather, information is processed collectively, in parallel throughout a network of nodes (the nodes, in this case, being neurons). The most common application of neural networks in computing today is “easy-for-a-human, difficult-for-a-machine” tasks, often referred to as pattern recognition. Applications range from optical character recognition (turning printed or handwritten scans into digital text) to facial recognition. One of the key elements of a neural network is its ability to *learn*. A neural network is not just a complex system, but a complex *adaptive* system, meaning it can change its internal structure based on the information flowing through it. Typically, this is achieved through the adjusting of *weights*. In the diagram above, each line represents a connection between two neurons and indicates the pathway for the flow of information. Each connection has a *weight*, a number that controls the signal between the two neurons. If the network generates a “good” output, there is no need to adjust the weights. However, if the network generates a “poor” output—an error, so to speak—then the system adapts, altering the weights in order to improve subsequent results.

There are several strategies for learning, and we'll examine two of them in this chapter.

- **Supervised Learning** - Essentially, a strategy that involves a teacher that is smarter than the network itself. For example, let its take the facial recognition example. The teacher shows the network a bunch of faces, and the teacher already knows the name associated with each face. The network makes its guesses, then the teacher provides the network with the answers. The network can then compare its answers to the known "correct" ones and make adjustments according to its errors
- **Unsupervised Learning** - Required when there is not an example data set with known answers. Imagine searching for a hidden pattern in a data set. An application of this is clustering, i.e. dividing a set of elements into groups according to some unknown pattern.
- **Reinforcement Learning** - A strategy built on observation. Think of a little mouse running through a maze. If it turns left, it gets a piece of cheese; if it turns right, it receives a little shock. The mouse will learn over time to turn left. Its neural network makes a decision with an outcome (turn left or right) and observes its environment (yes or not). If the observation is negative, the network can adjust its weights in order to make a different decision the next time.

This ability of a neural network to learn, to make adjustments to its structure over time, is what makes it so useful in the field of artificial intelligence. Here are some standard uses of neural networks in software today.

- **Pattern Recognition** - We have mentioned this several times already and it's probably the most common application. Examples are facial recognition, optical character recognition, etc.
- **Time Series Prediction** - Neural networks can be used to make predictions. Will the stock rise or fall tomorrow? Will it rain or be sunny?
- **Signal Processing** - Cochlear implants and hearing aids need to filter out unnecessary noise and amplify the important sounds. Neural networks can be trained to process an audio signal and filter it appropriately

3. Experiment

In this part of article we describe algorithm which serve to find handwritten number (Fig.1).

Recognize First Line

This level of algorithm separates handwritten characters from alphabet letters. We used here neural network called „Unsupervised learning“. That mean that neural network doesn't have input data vector with „right answers“. Input data was only vector consist from handwritten characters database and alphabet letters database. These data was processed in preprocessing block before. On the Fig.4a) we can see example of character p and on the Fig.3b) handwritten number 2 and on Fig.2a) we can see result of recognize. Neural network have only one output which can get values between 0 and 1. Value 1 (or near to 1) mean that input object is alphabet letter and result 0 (or near to zero) mean that object is handwritten number. After learning period neural network can reliably recognize handwritten numbers from alphabet characters. Disadvantage of this level is that noise (for example from scanned image), which preprocessing block cannot remove recognize neural network randomly as number or as letter. This disadvantage is remove in „recognize second line“.

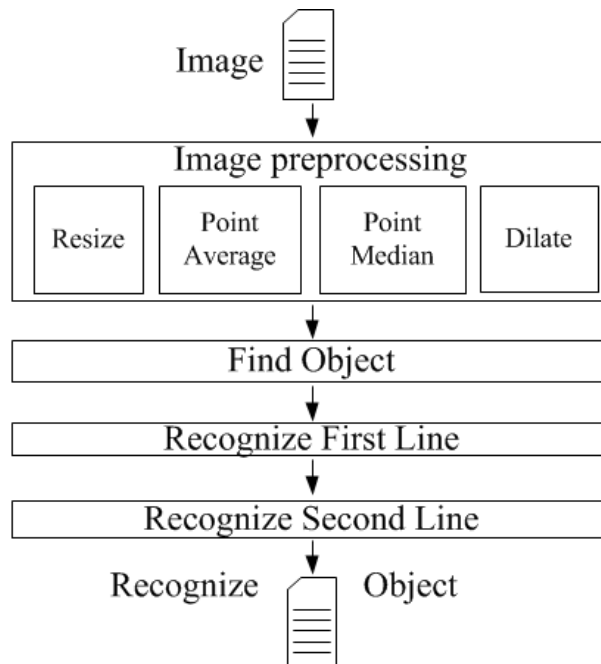


Fig.1: Object recognition flowchart

Recognize Second Line

On this level algorithm recognize directly handwritten numbers. On recognize we used neural network called „Supervised learning“. That mean that neural network had on input except vector which consist from handwritten letters also vector with „right answers“. Output from neural network is 10 numbers which can get values between 0 and 1. Position of output with highest value specifies the input number Fig.2b) Fig.3b). As we can see on Fig.2b) difference between highest and lowest value is significant. But that is not true when noise is on input (which was not remove in previous blocks) Fig.3a) Fig.4b). In this way we can divides noise from numbers in this block.

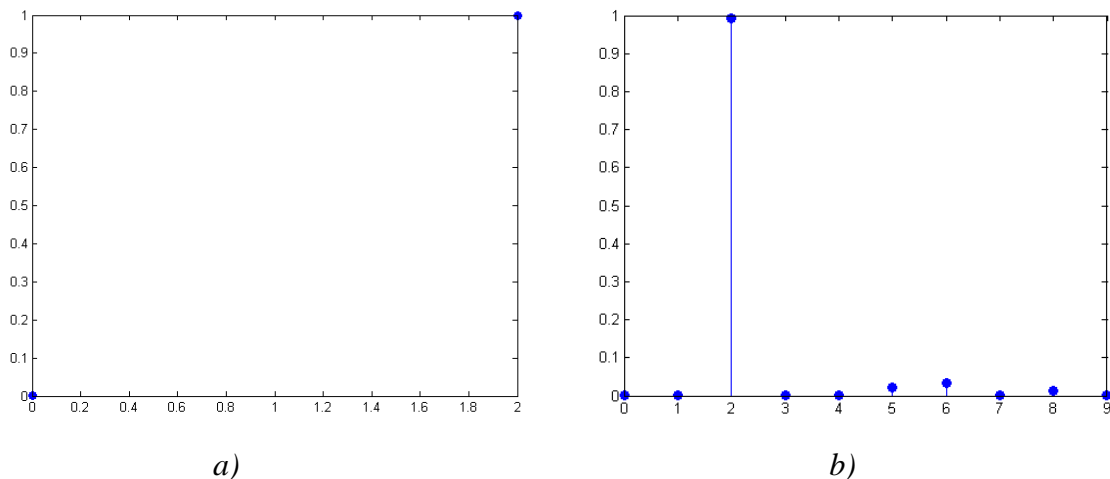
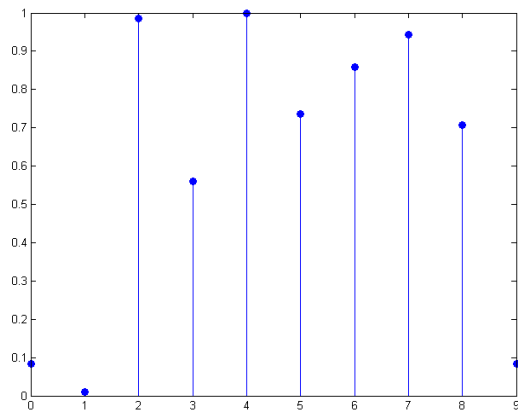


Fig. 2: a) Result on first line neural network output from handwritten number 2 and character letter p. b) Result on second line neural network output from number 2 on input



a)

b)

Fig. 3: a) Result on second line neural network output from noise on input.
b) Handwritten number 2.

Image preprocessing

At the start of recognition process is important remove noise from input image. Preprocessing block, in first step resize image. After that image passes through „Point Average“ and „Point Median“ blocks. These three blocks serve to reduce noise. At the end, dilatation block remove noise.



a)

b)

Fig. 4: a) Alphabet letter p b) Noise on input of second line neural network

Find object

In „find object“ block „find borders“ algorithm search all object in image. After searching all objects are resized to 56x56 pixel image and changed to 3136x1 vector which is input to neural network.

4. Conclusion

This paper focuses on handwritten numbers recognition in image. At the first step we created database 3690 handwritten numbers and alphabet characters. First part of algorithm removes noise from image and searching object in this image. Second part (neural network) divides objects to numbers and characters. Last part recognize numbers. Disadvantage of this algorithm is that in initialization process we have to set up several parameters (primary in preprocessing block) but values of these parameters is also depend on resolution changes and also on to difference between numbers size.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas ut orci neque. Nunc commodo

vulputate luctus. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia

Curae; 1 2 Nunc ligula neque, viverra vitae nulla at, vehicula elementum leo. Praesent

blandit id libero ut laoreet. Integer sit amet magna mollis, elementum ex sed, dapibus velit. Nam nec

Fig. 5: *Part of input image*

Acknowledgement

This work is resulting from the project VEGA 1/0987/12 sponsored by Ministry of Education, Slovak Republic.

References:

- [1] M. S. Nixon, A. S. Aguado, *Feature Extraction and Image Processing – Second edition*. Pearson Education. Oxford : Academic press, 2008.
- [2] S. Theodoridis, S. Koutroumbas, K *Pattern Recognition, Fourth Edition*. Boston: Academic press, 2009.
- [3] F. Y. Shih, *Image processing and Pattern Recognition - Fundamentals and Techniques*. New Jersey : John Wiley & Sons, 2010
- [4] P. Sinčák, G. Andrejková, *Neurónové siete - inžiniersky prístup - 1.diel : Dopredné siete*. Košice : Elfa, 1996.
- [5] S. Theodoridis, S. Koutroumbas, K *An Introduction to Pattern Recognition: A MATLAB Approach*. Oxford : Academic press.