

ARTIFICIAL NEURAL NETWORK FOR IDENTIFICATION OF LARGE LETTERS OF THE ALPHABET

BERK, P.; VINDIS, P.; LAKOTA, M.; OCEPEK, M. & RAKUN, J.

Abstract: One of the classical artificial neural network (ANN) applications is a system for character recognition. This system is the basis for different types of applications in various institutions, many applications are used in our everyday lives. Therefore, in our application we decided for the synthesis of ANN through the use of backpropagation method with adaptive learning rate and boot function. Character recognition system we developed using Matlab 2008b software tool. Each letter of the alphabet were presented with an array of dimensions $[20 \times 20]$ in the system and created a neural network algorithm is a matrix with a binary input code size $[20 \times 20]$. The result of processing ANN is a useful way of logical value 1, for successfully recognized symbol (letter X).

Key words: artificial neural network, synthesis, identification, start-up function



Authors' data: Bs. Elect. Eng. **Berk**, P[eter]; Dr. **Vindis**, P[eter]; Assoc. Prof. **Lakota**, M[iran]; Msc. **Ocepek**, M[arko]; Dr. **Rakun**, J[urij], University of Maribor, Faculty of Agriculture and Life Sciences, Pivola 10, SI-2311, Hoce, Slovenia, peter.berk@uni-mb.si, peter.vindis@uni-mb.si, miran.lakota@uni-mb.si, marko.ocepek@uni-mb.si, jurij.rakun@uni-mb.si

This Publication has to be referred as: Berk, P[eter]; Vindis, P[eter]; Lakota, M[iran]; Ocepek, M[arko] & Rakun, J[urij] (2011) Artificial Neural Network for Identification of Large Letters of the Alphabet, Chapter 31 in DAAAM International Scientific Book 2011, pp. 401-408, B. Katalinic (Ed.), Published by DAAAM International, ISBN 978-3-901509-84-1, ISSN 1726-9687, Vienna, Austria DOI: 10.2507/daaam.scibook.2011.31

1. Introduction

System with ANN is the basis for applications in institutions such as post offices, banks, security systems employment of robots (Abhijit et al., 1995). The review of checks, eye check at the entrance to the airport, when we teach a robot to pick up the item and then use the application for character recognition using neural networks (Sing, 2002). Savings are in cost and in less time spent on individual work.

The area which covers the identification of signs is called optical character recognition (Optical Character Recognition – OCR). The OCR application is these days used in banks, post offices, airports, business, agriculture (Stajnko et al., 2004). Readers should address sorting incoming and outgoing mail, reader checks in banks find similar checks, airline tickets and passport readers are used for various purposes such as finding the passenger's property, when checking the database in the reader through the required reading on 5800 models hour. OCR software is used in identifying and faxing, which allows the user to return to the graphic image of text document recognition. Identification of the eye, face, and handwriting is used in high security units. In many different areas this equipment has been started to use (Vindis et al, 2010). With the robotic device transparent we can read errors in the system, discover the edges, shapes and colours (Berk et al., 2010).

Optical system used for identifying is still a relatively unexplored field (Handwritten Recognition), which is based on simple character recognition system (Sadaoki, 2000). New ideas in computer industry, such as Microsoft Tablet PC is, a pen-based computation instead of a longer computational run-recognition system, which runs in the background instead if the real time.

2. Artificial neural network architecture using software and hardware components

ANN architecture (Figure 1) is based on the principle of the input matrix, consisting of 400 elements and 26 input neurons in the output layer to identify the characters (letters of the alphabet). ANN is composed of two layers (hidden layer, output layer). Log-sigmoid transfer function was chosen because it is obtained at the output of this function the value 0 to 1 and is handy for learning the output logical values (Abhijit et al., 1995). The hidden (first) layer is composed of 10 neurons. This number was chosen taking the guesswork and experience, in the event of a significant number of neurons in the hidden layer is a software package fault (not enough storage space in Matlab). If the ANN has difficulty in learning, we increase the number of neurons in the hidden layer.

A ANN has learned so that the output of networks represents a logical value of the correct classification of the output vector, while all other vectors at the output of ANN are set to a logical 0. In case of garbled characters (noise), the output of the ANN did not get the exact value of 1 or 0. After the ANN learned to recognize the character (letter X), we get the output of ANN as a logical value in the output vector of matrix [26x1].

Using this logic, we ensure that the network output matches the selected symbol (X), like this would happen in case of input noise vector. The result of

processing ANN is a useful way of logical value 1 for each recognized character, and all other elements in the matrix $[26 \times 1]$ take a value of 0.

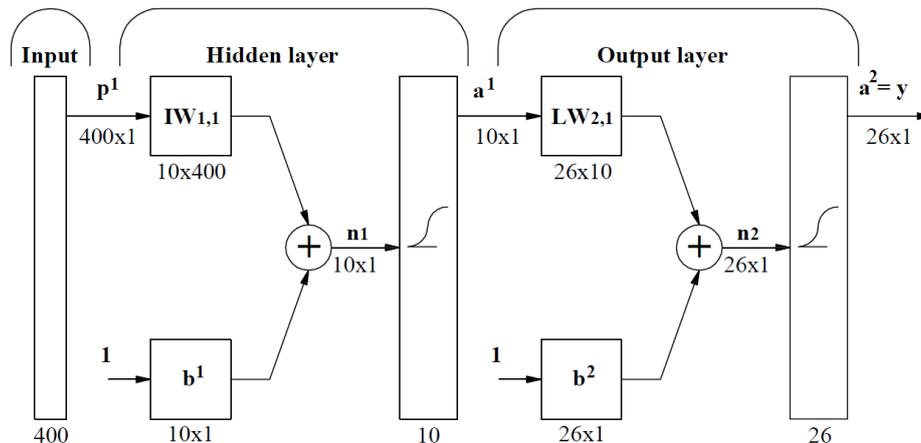


Fig. 1. ANN architecture

2.1 Hardware

Hardware consists of the HP Compaq 6830s Notebook NA779ES. A HP 6830s Laptop has a 3 GB DDR2 800 MHz memory, expandable to 8 GB. Hard Drive has capacity of 320 GB, supports SMART SATA bus, communicates and operates plants in 5400 in a minute.

2.2 Software Matlab 2008b

Matlab is a software package designed to numerical calculations, such as arithmetic operations vectors, difference equations and presentation of results. For such extensive use Matlab has got available libraries, through which we can come to more knowledge in the areas of neural network and designing filters. By entering commands in Matlab's workspace the programmer can directly input command through the command line in the command window or it can be commanded by entering into a separate file with the extension *.m. By recording command in the Command window or in a file the implementation can be repeated without re-entering. Matlab's functions are easily completed by a user through the Matlab programming language. The basic program structure is a matrix. In the Matlab program it is necessary to define variables and to reserve space in a memory. Effective programming in Matlab is achieved by pre-booking of memory and using commands with a dot (operate on the matrix components) and avoid use for loops.

3. Recognition letter system (methods)

We have created character recognition system through the simple steps in the software package Matlab. Each letter of the alphabet was presented with a character array of size $[20 \times 20]$, in the system of ANN.

Character array is composed of black and white pixels, a vector of logical value of 1 represents a black colour, a vector of logical value 0 and white. The vector we have constructed is shown in Figure 2 below.



Fig. 2. Character array

In a system for character recognition, we first made patterns of characters, with Adobe PDF and imported it into the MATLAB software package (Figure 2). We decided that we will make a system to identify large letters. The first problem that occurred was that an array size of the imported symbol was too big. In the case of over-sized array, we should take into account the fact that the PC processor speed increased every year for three times the value, we have the problem of a lack of CPU power to operate the system in real time. Studying ANN can take days and to find the final result may take hours and hours. Therefore, we chose a character array of size [20x20] pixels.

A ANN (Figure 1) receives its input 400 value (logic value 0 or 1) as the input vector with a matrix size [400 x 1]. From neural network algorithm is then required to identify the character (letter) of the output value of the vector with 26 elements. Output vector, which consists of 26 elements, each represents a character (letter) in the alphabet. For the proper functioning of the algorithm, neural network takes the output logical value 1, where it is perceived the correct character (letter). All other values of the output vector take a logical value 0. A network works well if the input character (input vector) flawed, in practice very often we have the case that a sign is a little distorted (containing noise).

3.1 Setting the weights of the ANN

In our case we have an ANN with two weights; weight in the hidden layer and weights in the output layer (Figure 1). These weights represent the memory of neural networks, where end-trained weights are used when the network is in operation (Abhijit et al., 1995). Initial weights generate at random, then, the updated weights serve as an error (difference) between the actual output value of network and desired (target) output.

In the case of the occurrence of updating the weights at each repetition, the neural network then learns, because the repetition updates values of the weights are going for so long, until a minimum error network. First we were defining the notion of the samples stored in shaped, and consist of binary values 0 and 1. The added layers of weights can be increased, but the added layer is then impossible to be used. The received input from the left side of each incoming and reciprocal links is associated with weight (w_{ji}). Perceived procedural execution is the sum of input values which are given in the equation below:

$$\text{net} = \sum_{i=1}^n O_i \cdot W_i \quad (1)$$

Where:

O_i - output layer neurons

W_i - weights

United weights each time the connection is adjusted during the learning ANN after completing the learning network weights occupy a fixed value from 0 to 1. Below is given an array with the values of the weights that corresponds to each of the connection. This matrix is represented by a matrix of dimensions [20 x 20], Figure 3, each cell of the matrix has different values from 0 to 1 and is used in various layers of ANN.

Columns 1 through 17																			
0.2162	0.3403	0.0920	0.5926	0.3473	0.7777	0.8784	0.1348	0.9432	0.3518	0.9403	0.9583	0.0778	0.5858	0.9851	0.2428	0.1139			
0.6396	0.0290	0.9823	0.7198	0.9901	0.4049	0.9821	0.4594	0.5368	0.0862	0.9861	0.4763	0.9883	0.6651	0.2693	0.9397	0.4185			
0.4390	0.7043	0.3479	0.2494	0.6037	0.1747	0.9386	0.3829	0.3285	0.0948	0.1038	0.2251	0.3416	0.6845	0.4091	0.1393	0.6809			
0.2611	0.7308	0.3967	0.5501	0.0041	0.3055	0.5000	0.8313	0.9559	0.5150	0.5635	0.4977	0.1535	0.6877	0.2199	0.0911	0.4495			
0.8606	0.0879	0.6240	0.3522	0.9918	0.8039	0.5757	0.3018	0.8233	0.3096	0.4393	0.4034	0.7087	0.4867	0.7895	0.0960	0.3553			
0.0695	0.0116	0.5182	0.9510	0.6769	0.9141	0.5067	0.8436	0.9540	0.6334	0.5378	0.7572	0.2801	0.2628	0.5322	0.7969	0.5886			
0.5330	0.7029	0.3956	0.1030	0.6755	0.8613	0.9498	0.9451	0.4146	0.1757	0.7568	0.5862	0.5942	0.2017	0.3100	0.1389	0.4502			
0.8956	0.5008	0.9697	0.1199	0.0233	0.9264	0.0393	0.0040	0.1328	0.3771	0.4013	0.4260	0.2401	0.4845	0.6615	0.1026	0.7028			
0.5550	0.2404	0.2618	0.5067	0.8207	0.8035	0.1786	0.6344	0.4938	0.8253	0.3064	0.9582	0.9370	0.9918	0.1076	0.4773	0.0018			
0.8717	0.5237	0.1874	0.3708	0.3024	0.4148	0.6537	0.9148	0.3494	0.5232	0.5484	0.0414	0.1491	0.4530	0.1963	0.0817	0.5323			
0.4752	0.4186	0.1197	0.1012	0.1501	0.3097	0.1896	0.0067	0.0523	0.2520	0.4146	0.2123	0.4965	0.9254	0.1487	0.5926	0.5405			
0.7183	0.2834	0.6278	0.2270	0.2798	0.3431	0.0894	0.1273	0.7624	0.2196	0.5288	0.4445	0.7609	0.4591	0.2616	0.1475	0.7427			
0.5454	0.6463	0.9633	0.5785	0.3332	0.4905	0.4463	0.7225	0.4912	0.8734	0.2915	0.7015	0.9510	0.3220	0.8307	0.9067	0.8894			
0.2460	0.0438	0.5558	0.5016	0.3027	0.8449	0.3045	0.8048	0.1681	0.1366	0.4874	0.4419	0.2413	0.8290	0.1179	0.3493	0.2515			
0.7853	0.3458	0.3082	0.2876	0.5181	0.1028	0.4764	0.8256	0.3312	0.0170	0.8777	0.5411	0.6874	0.5933	0.2547	0.2980	0.7674			
0.4850	0.9800	0.8948	0.5320	0.7887	0.5633	0.3493	0.3282	0.8030	0.6963	0.2362	0.4727	0.0163	0.1148	0.7473	0.4540	0.1276			
0.5117	0.0831	0.3161	0.8390	0.3375	0.2533	0.3720	0.3249	0.1671	0.5379	0.7172	0.8289	0.9145	0.0827	0.9976	0.6859	0.0381			
0.9366	0.3717	0.4831	0.3493	0.5915	0.1217	0.8864	0.1537	0.5839	0.2036	0.3105	0.8846	0.9043	0.4074	0.1463	0.0358	0.4047			
0.3777	0.8699	0.0891	0.6559	0.5653	0.1057	0.3899	0.2855	0.0380	0.0264	0.7090	0.0294	0.3271	0.2548	0.6625	0.6721	0.1735			
0.1555	0.4788	0.8536	0.7512	0.2224	0.6954	0.1917	0.4096	0.3960	0.9026	0.0609	0.2871	0.8724	0.7509	0.1267	0.7323	0.8141			
Columns 18 through 20																			
0.4909	0.2128	0.7674																	
0.4039	0.5311	0.6368																	
0.9251	0.0564	0.0217																	
0.7811	0.4674	0.0915																	
0.6558	0.0404	0.7790																	
0.8952	0.8794	0.0288																	
0.6417	0.9492	0.1157																	
0.4082	0.0394	0.0687																	
0.2785	0.8670	0.3726																	
0.0094	0.6287	0.1220																	
0.2111	0.7772	0.0420																	
0.4557	0.4689	0.5905																	
0.7951	0.9704	0.9810																	
0.1063	0.7408	0.9278																	
0.6771	0.6455	0.2600																	
0.1673	0.9663	0.2305																	
0.4704	0.5554	0.9342																	
0.0542	0.6838	0.9134																	
0.6379	0.9587	0.9980																	
0.2390	0.3189	0.6655																	

Fig. 3. Character matrix [20 x 20] with the values of weights from 0 to 1

3.2 Learning ANN

To make an ANN which will recognize the characters is not perfect, but contains noise, we learned network with ideal data and the data that contains noise. To do this, we first learn network with an ideal input data, until the minimum allowable deviations from the ideal value, in our case is the ideal value of 1. Then ANN learns both, the ideal input data and the data noise. A network learns in two directions, network learns the ideal input data, while the neural network learns the data 2 consists of a noise (Figure 8). Two-way learning ability of neural networks maintain an ideal distribution of the input vector (a matrix of 400 x 1). Full learning with neural network is done through the use of backpropagation method with adaptive learning rate and start-up function, which we used in the software package Matlab 2008b and called **trainbpx** (Charalambous, 1992).

3.3 Initializing the input and output signal

The input signal is initialized when it appears on the entrance sign of ANN (pattern with the letter, in this case of a matrix of size [400 x 1], with binary values from 0 to 1), each input device at the entrance network model corresponds to the letter of the alphabet and each unit takes a certain amount of input. Each output of hidden neuron is then calculated using the sigmoid function. Sigmoid function equation is:

$$f(x) = \frac{1}{1+e^{\sum_{i=1}^n O_i \cdot W_i}} \quad (2)$$

Where:

O_i - output layer neurons

W_i - weights

a.) Input layer (i)

Input layer consists of 26 inputs, which are stored in the file (DAT). The input layer neurons = output layer neuron ($I_i = O_i$).

b.) Hidden layer (h)

Input hidden layer is given by the equation:

$$h = I_k = \sum_i^n W_{ki} \cdot O_i \quad (3)$$

Where:

O_i - output layer neurons

W_{ki} - weights

where we take values on our entrance to the equation:

$$A \cdot I \cdot S \quad (4)$$

Where:

A - input matrix

I - array of hidden layer

S - matrix of the sigmoid function (Figure 4)

Columns 1 through 17

0.5562	0.4799	0.3286	0.1836	0.5360	0.9534	0.6998	0.5486	0.7858	0.3749	0.1476	0.5613	0.8580	0.3469	0.9848	0.3700	0.7955
0.8504	0.9582	0.7538	0.4579	0.0629	0.5233	0.5824	0.3399	0.1955	0.3916	0.1632	0.9617	0.2086	0.1840	0.2917	0.7723	0.0594
0.7281	0.0857	0.1967	0.0447	0.2152	0.9378	0.8670	0.8848	0.3931	0.5620	0.0619	0.7629	0.8585	0.9009	0.9532	0.0956	0.1558
0.2421	0.9375	0.3097	0.5528	0.6623	0.4434	0.3462	0.7012	0.2294	0.9791	0.3585	0.0608	0.0308	0.5771	0.4590	0.1446	0.6367
0.7249	0.7828	0.1476	0.9216	0.2233	0.0845	0.7793	0.9835	0.9735	0.6329	0.9862	0.6080	0.0460	0.2280	0.4490	0.0941	0.5490
0.2398	0.4012	0.6505	0.5603	0.6483	0.2345	0.1780	0.9496	0.9803	0.7843	0.7333	0.1847	0.5378	0.0097	0.2397	0.6036	0.8743
0.1998	0.0478	0.2185	0.5002	0.1164	0.7581	0.5296	0.8163	0.9944	0.9390	0.7321	0.4945	0.4894	0.6459	0.7834	0.2472	0.2065
0.4640	0.1747	0.2398	0.7085	0.1112	0.4308	0.3569	0.4787	0.6412	0.1361	0.7987	0.2503	0.3588	0.7239	0.8395	0.7434	0.1318
0.4600	0.9167	0.1614	0.9378	0.8744	0.6003	0.2574	0.5882	0.7855	0.6549	0.8167	0.4325	0.3141	0.3122	0.4573	0.3366	0.7183
0.4625	0.6320	0.9787	0.1261	0.3189	0.5420	0.7063	0.2271	0.0352	0.7493	0.6470	0.5063	0.3913	0.5806	0.4213	0.7603	0.2834
0.4613	0.3745	0.7126	0.9683	0.9819	0.7190	0.7657	0.4481	0.3183	0.7756	0.1030	0.0542	0.2332	0.9994	0.6414	0.8314	0.6914
0.6711	0.3187	0.3894	0.4924	0.9751	0.5388	0.5955	0.5492	0.0886	0.0857	0.5262	0.1796	0.6462	0.0482	0.7152	0.7496	0.1931
0.2303	0.0089	0.0795	0.1922	0.4015	0.7603	0.4538	0.6292	0.6020	0.3292	0.2774	0.4404	0.1537	0.2798	0.8143	0.3226	0.6044
0.4945	0.8134	0.1362	0.5950	0.9633	0.9608	0.9702	0.5319	0.4123	0.5404	0.7146	0.3596	0.3413	0.3659	0.0315	0.8056	0.6338
0.8707	0.5094	0.7253	0.7727	0.1118	0.5681	0.8264	0.5449	0.6786	0.5716	0.2285	0.7153	0.8524	0.3579	0.5031	0.4286	0.5204
0.4410	0.6627	0.0585	0.5511	0.3203	0.3938	0.2000	0.5346	0.3501	0.8780	0.1365	0.8272	0.9134	0.3863	0.8802	0.5103	0.3074
0.5585	0.9839	0.5971	0.1704	0.9970	0.1022	0.3762	0.2599	0.0024	0.0357	0.2584	0.8353	0.9925	0.6957	0.4557	0.1433	0.3407
0.0239	0.9215	0.9669	0.7979	0.6620	0.7180	0.4316	0.1677	0.9977	0.0119	0.2199	0.7669	0.3318	0.7313	0.9792	0.9252	0.0641
0.5651	0.2045	0.8561	0.0766	0.4097	0.1505	0.9114	0.7317	0.9418	0.1960	0.5385	0.8539	0.0594	0.4004	0.7227	0.4602	0.6695
0.3204	0.3360	0.4998	0.8711	0.2113	0.2602	0.9642	0.6496	0.5826	0.3629	0.6402	0.8258	0.0487	0.6813	0.7016	0.2639	0.6929

Columns 18 through 20

0.3949	0.4306	0.9730
0.5132	0.6880	0.4660
0.4862	0.3015	0.3895
0.0255	0.0011	0.3662
0.3169	0.0085	0.1218
0.3560	0.9347	0.3978
0.9329	0.1153	0.8388
0.2732	0.2545	0.9482
0.2847	0.8709	0.5814
0.3565	0.4138	0.7036
0.2075	0.7640	0.2448
0.8498	0.2747	0.9452
0.8673	0.1577	0.2727
0.5560	0.2842	0.4877
0.8409	0.2502	0.5123
0.5445	0.1025	0.2974
0.4839	0.3058	0.4736
0.6370	0.3901	0.0097
0.7920	0.7027	0.4455
0.8037	0.2660	0.1347

Fig. 4. An array of sigmoid function

Output of hidden layer is calculated by the equation:

$$h = O_h = \frac{1}{1+e^{-I_h}} \tag{5}$$

Where:

I_h - final value

4. Results

Reliability of ANN were determined by testing neural network with some 100 elements of the input vector by varying noise. ANN was tested using a script file appcr1.mv (software package Matlab 2008b), at different levels of noise, the result is presented in a graph showing the deviation in % for the neural network 1 (ideal sign at the entrance) and neural network 2 (the ideal character add noise at the input neural network), Figure 8. For noise, we predict a constant value in the range of 0.05 to 0.5, these values we then add to the signal at the entrance of a network. At each noise level were made 10 different presentations noisy versions of each character and for all these versions were calculated using a script file appcr1.m (software package Matlab 2008b). Output ANN has had to match the correlation transfer function, which provides that only one of 26 outputs (letter A - X) of networks reach a logical value of 1. In our case, we learned network at different levels of noise vector, where we forced neural network to this, that has learned to act with regard to noise, where an obvious problem in the real world. Figure 5, 6 and 7 below show the results of the identification of ideal characters X and character X with noise, using artificial neural networks.

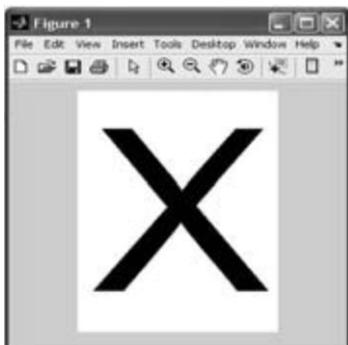


Fig. 5. Ideal character, input of ANN



Fig. 6. Character with noise, input of ANN



Fig. 7: Recognition character, output of ANN

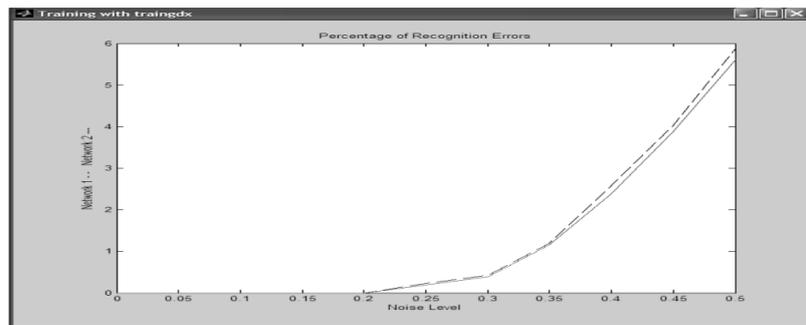


Fig. 8. Deviation in % of input noise level, the neural network 1 and 2

5. Conclusions

Example applications of ANN (recognition characters), shows how you can construct a simple sample application for character recognition. ANN is taught in our case a few times at different values of the input vector. The disadvantage of networks is that they do not have a mechanism through which these solutions could be explained. Therefore, they are less suitable for use in deciding successful when used to identify patterns. In the future we want to do with the help of ANN identification forms of maize from the image, using the identified model could then activate the actuators for selective coating product.

6. References

- Abhijit, S. Pandya & Robert B. Macy (1995). *Pattern Recognition with Neural Networks in C++*, ISBN: 9780849394621, Florida Atlantic University, Boca Raton, Florida, USA
- Berk, P.; Rakun, J.; Vindis, P.; Stajanko, D.; & Miran, L. (2010). *Temperature loop structure with fuzzy and classical controllers*, ISBN 978-3-901509-84-1, V: Katalinić, Branko (ur.), DAAAM International scientific book 2010, Vienna: DAAAM International Publishing, pp 117-128
- Sadaoki, F. (2000). *Digital Speech Processing: Synthesis, and Recognition, Second Edition*, ISBN: 9780824704520, Tokyo Institute of Technology, Tokyo, Japan
- Sing, T. Bow (2002). *Pattern Recognition and Image Preprocessing*, ISBN: 9780824706593, Northern Illinois University, Dekalb, USA
- Stajanko, D.; Lakota, M. & Hocevar, M. (2004). *Estimation of number and diameter of apple fruits in an orchard during the growing season by thermal imaging*, Comput. electron. agric., [Print ed.], vol. 42, pp 31-42
- Vindis, P.; Stajanko, D.; Berk, P. & Lakota, M. (2010). *Qualitative multi-criteria evaluation of energy crops*, ISBN 978-3-901509-84-1, V: KATALINIĆ, Branko (ur.), DAAAM International scientific book 2010, Vienna: DAAAM International Publishing, pp 99-116
- Charalambous, C. (1992). *Conjugate gradient algorithm for efficient training of artificial neural networks*, ISSN: 0956-3768, Cyprus Inst. of Neurology & Genetics, Nicosia, vol. 139, no. 3, pp 301-310
- *** (2001), http://courses.washington.edu/css457/matlab/learning_matlab.pdf - The Mathworks, MATLAB student version, Accessed: 2011-05-04