Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IMPACT FACTOR: 6.017

IJCSMC, Vol. 8, Issue. 2, February 2019, pg.20 – 33

Performance Analysis of Artificial Neural Networks used for Color Image Recognition and Retrieving

Jamil Al-Azzeh[1], Ziad Alqadi[2], Mohammed Abuzalata[3]

[1]Computer Engineering Department, Al Balqa'a Applied University, Amman, 11134, Jordan
E-mail: azzehjamil@gmail.com
[2]Computer Engineering Department, Al Balqa'a Applied University, Amman, 11134, Jordan
E-mail: Natalia_maw@yahoo.com
[3]Computer Engineering Department, Al Balqa'a Applied University, Amman, 11134, Jordan
E-mail: abuzalata@bau.edu.jo

Abstract: Image recognition refers to methods and technologies that identify digital images such as places, logos, people, objects, buildings, and several other variables in images. Users are sharing vast amounts of data through applications, social networks, and websites. The large volume of digital data is being used by companies to deliver better and smarter services to the people accessing it.

In this paper we will introduce a methodology of image recognition and retrieval process based on using ANN as a recognizer capable to identify digital image index. ANN will be implemented and tested in order to select an optimal ANN which must give the accurate image index with minimum MSE and maximum PSNR.

Keywords: Features, ANN, activation function, training function, MSE, PSNR.

1- Introduction

1-1 Artificial Neural Network

Artificial neural networks (ANN) [1-35] is a set of fully connected neurons as shown in figure (1), each neuron has a input and a set of weights, the output is generated by applying the summation process and according to the selected activation function the output is generated.

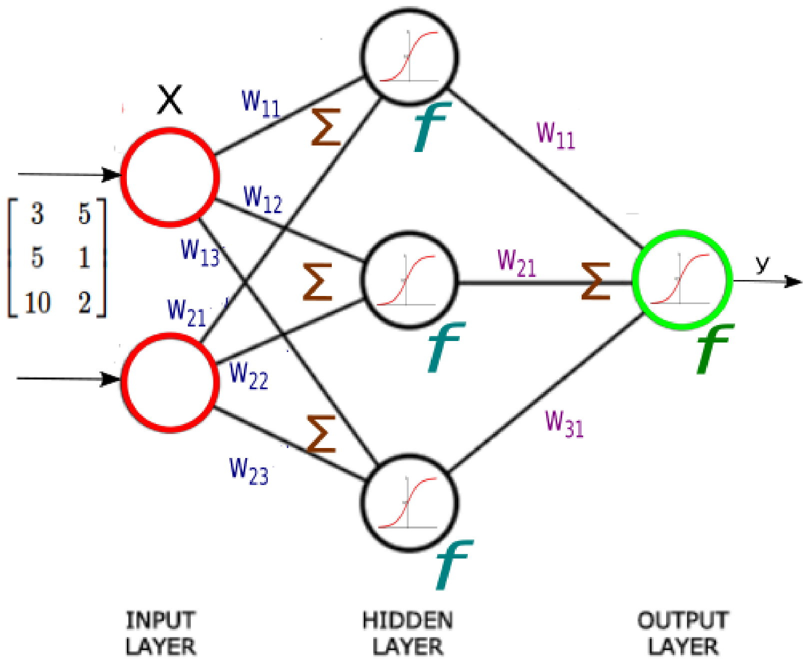

Figure (1): ANN sample example

ANN is an important mathematical tool, and it is applicable for various industrial and engineering applications, such as color image recognition and retrieval [2], [3].

To maximize the efficiency of the used ANN (Getting the calculated output with zero errors between the target and the calculated output) and minimum processing time) we have to consider the following factors [4], [5]:

- Defining the input data set.
- What are the calculated targets to be achieved?
- Selecting ANN architecture (How many layers, and how many neurons in each layer).
- Selecting the training function.
- For each layer what is the activation function to be used? (Logsig, tansig, or linear).
- What is the acceptable error value? (To be closed to zero).

- How many training cycles to be used ?(one training cycle includes a forward phase to calculate the neuron outputs, checking the error, if the error is not acceptable process the back word process to adjust the neurons weights).

- Select a starting values for the weights (easily is to initialize all the weights to zero).

  Taking the above factors into consideration, using ANN as a recognition or retrieval tool we have to apply the following tasks:

  - ANN creation

  1) Setting the input data set.
  2) Setting the targets.
  3) Creating ANN using a selected architecture.
  4) Setting all the weights to zero.
  5) Selecting some ANN parameters (Goal (error) =0, Number of training cycles to be used (epochs)).
  6) Raining ANN.
  7) If the error equal zero, save ANN to be used for recognition, otherwise adjust the number of cycles, or adjust ANN architecture and repeat this phase.

  - ANN processing

    This phase is being used for image recognition or retraining and it can be implemented applying the following steps:

    1) Load ANN.
    2) Select color image features to be used as an input.
    3) Get the image index by running ANN.
    4) Use the index to recognize or retrieve the image from the images library or data base.

1-2 Color image features.

Digital color image is a 3D matrix as shown in figure (2), the first matrix represents the red color, and the second one represents the green color, while the third one represents the blue color[5], [6].

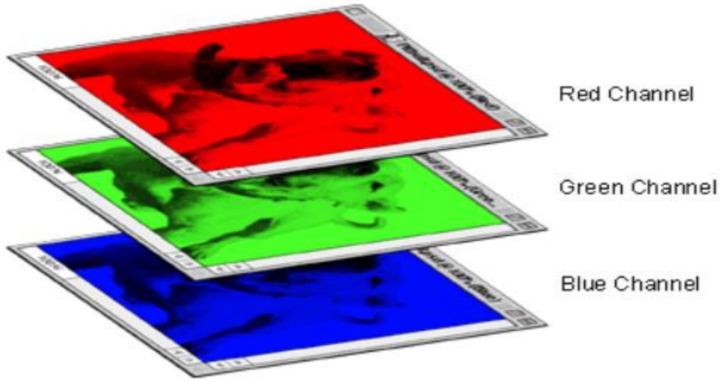

Figure (2): 3D color image

Color image usually has a huge size, which makes it difficult to retrieve or recognize the image pixel by pixel, for example the image shown in figure (3) has a size of 384x512x3=589824 pixel, this image requires 0.075000 seconds to be recognized and this time is high and must be reduced in order to achieve efficient method of recognition.

Figure (3): Color image example

To enhance the process of image recognition or retrieval we can use a method to extract small number of values, which form an image vector of features, then this vector can be passed to ANN to get the image index, which can be easily used to recognize or retrieve the image as shown in figure (4).

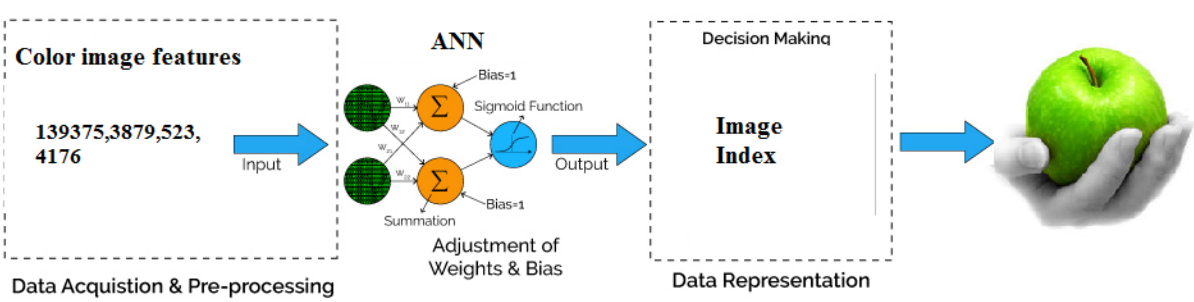

Figure (4): Recognition process.

Many methods are now used based on local binary pattern (LBP) such as center symmetric LBP (CSLBP) [7], [8], reduced LBP [9]. Figure (5) illustrates how LBP operator for a pixel can be calculated:

In our experimental part we will use the following method of image features extraction [5], [10], and [11], which is shown in figure (6):

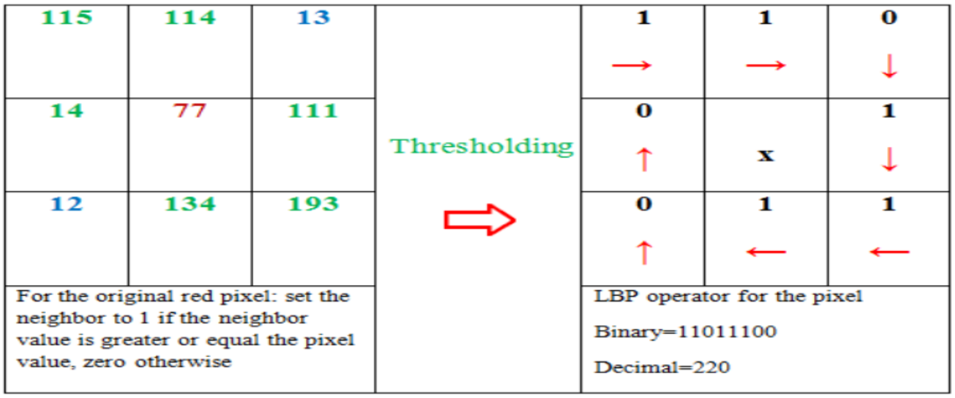

Figure (5): LBP calculation

P(i-2,j-2)	P(i-2,j-1)	P(i-2,j)	P(i-2,j+1)	P(i-2,j+2)
P(i-1,j-2)	P(i-1,j-1)	P(i-1,j)	P(i-1,j+1)	P(i-1,j+2)
P(i,j-2)	P(i,j-1)	P(i,j)	P(i,j+1)	P(i,j+2)
P(i+1,j-2)	P(i+1,j-1)	P(i+1,j)	P(i+1,j+1)	P(i+1,j+2)
P(i+2,j-2)	P(i+2,j-1)	P(i+2,j)	P(i+2,j+1)	P(i+2,j+2)

```
A1=( P(i-2,j-2)+P(i-2,j-1)+P(i-2,j)+P(i-2,j+1)+P(i-2,j+2)
       +P(i-1,j+2)+P(i,j+2)+P(i+1,j+2)+P(i+2,j+2)
       +P(i+2,j-2)+P(i+2,j-1)+P(i+2,j)+P(i+2,j+1)
       +P (i-1, j-2) +P (i, j-2) +P (i+1, j-2))/16
A0=( P(i-1,j-1)+P(i-1,j)+P(i-1,j+1)  +P(i+1,j-1)+P(i+1,j)
      +P (i+1, j+1) + P (i, j-1) +P (i, j+1))/8
If (A0>=P (i, j) then K0=1
Else
K0=0.
If (A1>=P (i, j) then K1=1
Else
K1=0.
F=K0+2*K1
Features (F+1) =Features (F+1) +1.
```

Figure (6): The used method to calculate LBP operator

2- Implementation and Experimental Results

2-1 Color Image Features

The above mentioned method of color image features was implemented using various color images in types and sizes; table (1) shows a sample of these features:

Table (1): Color images features

Image number	Features			
1	**139375**	**3879**	**523**	**4176**
2	**114516**	**18986**	**623**	**13371**
3	**3640245**	**61727**	**553**	**13107**
4	**88123**	**27788**	**2116**	**29973**
5	**132757**	**5034**	**863**	**9151**
6	**115642**	**12729**	**1751**	**17573**
7	**116967**	**9555**	**2276**	**19724**
8	**127722**	**4134**	**1483**	**14661**
9	**129888**	**9710**	**251**	**7541**
10	**53737**	**6410**	**3165**	**84654**

From table (1) we can see that the array vector for each image is a unique, thus it can be used as a key or signature to recognize or retrieve the image from images database.

The features in table (1) must be reorganized to form the input data set to ANN, by creating a 4 by n matrix (n number of images in the data base), then the data set must by divided by a big number in order to avoid receiving all the time the output 1 when applying tansig or logsig activation functions as shown in figures (7) and (8):

```
Columns 1 through 6

    139375       114516      3640245        88123       132757       115642
      3879        18986        61727        27788         5034        12729
       523          623          553         2116          863         1751
      4176        13371        13107        29973         9151        17573

Columns 7 through 10

    116967       127722       129888        53737
      9555         4134         9710         6410
      2276         1483          251         3165
     19724        14661         7541        84654
```

Figure (7): Arranging features matrix

```
Columns 1 through 8

  0.1394    0.1145    3.6402    0.0881    0.1328    0.1156    0.1170    0.1277
  0.0039    0.0190    0.0617    0.0278    0.0050    0.0127    0.0096    0.0041
  0.0005    0.0006    0.0006    0.0021    0.0009    0.0018    0.0023    0.0015
  0.0042    0.0134    0.0131    0.0300    0.0092    0.0176    0.0197    0.0147

Columns 9 through 10

  0.1299    0.0537
  0.0097    0.0064
  0.0003    0.0032
  0.0075    0.0847
```

Figure (8): Dividing features matrix by 1000000

2-2 ANN Implementation

In this part we will introduce procedures needed to create an optimal ANN used to recognize or retrieve color image using its features. Here optimality means creating ANN with the following features:

- Accuracy : Moving mean square error (MSE) between the target output(T) and the calculated output(image index)(CO) to zero or maximizing the value of peak signal to noise ratio (PSNR), these parameters can be calculated as shown in following equations:

$$MSE = \frac{\sum_{i=1}^{n} ABS(T(i) - CO(i)))^2}{n}$$

$$PSNR = 10 * log10((10.^2)/MSE)$$

$$n = numberofoutputs$$

- Minimum training time

To achieve the above mentioned parameters we have to be carefully when answering the following questions:

- How many layers in the required ANN?
- How many neurons in each layer?
- Which activation function to select for each layer? (for the output layer it must be linear and for other layers it can be logsig or tansig).
- Which training function to be used?

  We have to select one of the following training functions:

1) **Trainlm:** Levenberg-Marquardt back propagation training function.
2) **Traingd:** Gradient descent back propagation training function.
3) **Traingdm**: Gradient descent with momentum back propagation training function.

4) **Traingdx**: Gradient descent with momentum and adaptive learning rate back propagation training function.

- How many training cycles to be used for ANN training?
- What are the neurons weights? (Initially it is better to initialize them to zeros).
- What is the value of training rate? (Usually used to adjust the weights during the training process).

To answer the above questions we used the following matlab code several times varying ANN parameters:

```
clear all, clc
%Load input data
load a
idata=a/1000000; target=1:10;
tic
%Create ANN
netz=newff(minmax(idata),[4 1],{'tansig','purelin'},'trainlm');
%Initialize ANN paramaters
netz=init(netz); netz.trainParam.goal=0; netz.trainParam.epochs=20000;
netz.trainParam.lr=0.002;
%Train ANN
netz=train(netz,idata,target)
toc
%Run ANN
 y=(sim(netz,idata))
  MSE=0;
for i=1:10
MSE=MSE+((abs(target(1,i)-y(1,i))).^2);
end
MSE=MSE./10
PSNR=10*log10((10.^2) /MSE)
```

Experiment 1:

Fixing training rate to 0.2 and training cycles to 1000

Table (2) shows the results of the first experiment (Number of layer 2[input with 4 neurons and output layer with one neuron], training rate=0.2 maximum training cycles=1000 varying the training rate:

Table (2): Experiment 1 results

Training function	Activation function	Training cycles	MSE	PSNR	Training time(seconds)
trainlm	**tansig**	**42**	**8.3357e-027**	**280.7906**	**0.615000** **(optimal)**
trainlm	logsig	89	2.7067e-025	265.6757	0.771000
traingd	tansig	1000	4.8297	13.1608	2.673000
traingd	logsig	1000	4.4665	13.5003	2.353000
traingdm	tansig	1000	4.5299	13.4391	2.489000
traingdm	logsig	1000	4.6187	13.3548	2.419000
traingdx	tansig	1000	2.8331	15.4773	2.346000
traingdx	logsig	1000	3.8044	14.1971	2.422000

From table (2) we can see that the optimal ANN is with 2 layers and the activation function to be used for input layer is tansig, while the training function must be trainlm. Figure (9) shows MSE using tansig, while figure (10) shows MSE using logsig.

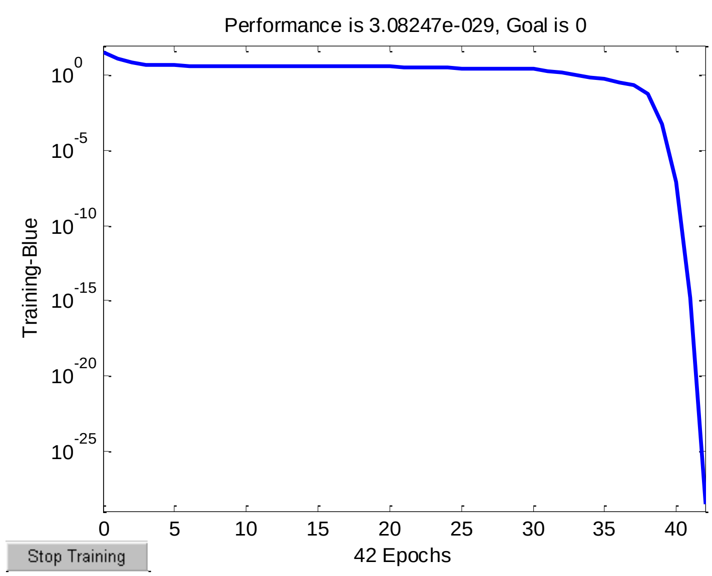

Figure (9) : MSE using tansig

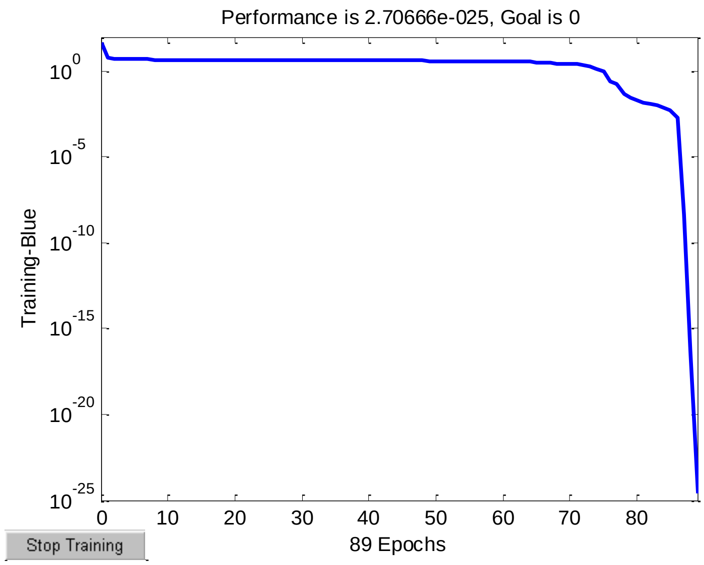

Figure (10): MSE using logsig

Experiment 2:

Fixing training rate to 0.2 and training cycles to 20000

Table (3) shows the results of this experiment

Number of layer 2[input with 4 neurons and output layer with one neuron] training rate=0.002 maximum training cycles=20000

Table (3): Experiment 2 results

Training function	Activation function	Training cycles	MSE	PSNR	Training time(seconds)
trainlm	tansig	120	4.3293e-028	280.7906	0.809000
trainlm	**logsig**	**235**	**2.7238e-026**	**293.6359**	**1.199000** **(optimal)**
traingd	tansig	20000	3.7796	14.2256	41.985000
traingd	logsig	20000	4.4637	13.5030	42.229000
traingdm	tansig	20000	4.4849	13.4824	43.083000
traingdm	logsig	20000	4.8532	13.1397	42.978000
traingdx	tansig	20000	1.0136	19.9414	47.332000
traingdx	logsig	20000	1.3099	18.8275	47.544000

Here we can see that logsig gives a optimal ANN

Experiment 3

Here we will use only trrainlm training function, because it is the best to be used, and we vary the number of layers:

Table (4) shows the results of this experiment (training rate =0.2)

From table (4) we can see that adding a new hidden layer to ANN architecture will improve ANN accuracy by increasing PSNR and decreasing MSE but with an extra time for training and extra space to save ANN.

Table (4): Experiment 3 results

L:logsig, T:tansig, LN: linear					
Layers	Activation function	Training time(seconds)	ANN size(bytes)	MSE	PSNR
4 1	L, LN	0.684000	26785	1.5399e-028	298.1252
4 1	T, LN	0.762000	26785	5.1087e-028	292.9169
4 4 1	L, L, LN	0.798000	33487	3.0545e-028	295.1506
4 4 1	L, T, LN	0.774000	33487	3.4699e-027	284.5969
4 4 1	**T, L, LN**	**0.611000**	**33487**	**7.5021e-029**	**301.2482** **(optimal)**
4 4 1	T, T, LN	0.715000	33487	1.0657e-023	249.7238
4 8 1	L, L, LN	0.643000	34095	8.4985e-023	240.7066
4 8 1	L, T, LN	0.849000	34095	8.5217e-029	300.6948
4 8 1	T, L, LN	0.648000	34095	1.7247e-022	237.6328
4 8 1	T, T, LN	0.777000	34095	2.0593e-025	266.8627

Conclusion

Color image features were extracted and passed to ANN. It was shown that selecting the following will lead to achieving optimal ANN by mean of maximizing:

- Using the training function trainlm improve ANN performance.
- It is preferable to use input layer with number of neurons equal to the number of inputs in the input data set.

- For hidden layers we have to use logsig or tansig activation function.
- Adding a hidden layer will improve ANN performance.

References

1. Akram A. Moustafa1, Ziad A. Alqadi and Eyad A. Shahroury, Performance Evaluation of Artificial Neural Networks for Spatial Data Analysis, WSEAS TRANSACTIONS on COMPUTERS, Issue 4, Volume 10, pp 115-124 April 2011.
2. Insung Jung, and Gi-Nam Wang, "Pattern Classification of Back-Propagation Algorithm Using Exclusive Connecting Network, World Academy of Science, Engineering and Technology, vol 36, 2007, pp189-193.
3. M.Hagan, H. Demuth, M. Beele, Neural Network Design, University of Colorado Bookstore, 2002, ISBN: 0- 9717321- 0-8.
4. Khaled M. Matrouk,Haitham A. Alasha'ary, Abdullah I. Al-Hasanat, Ziad A. Al-Qadi, Hasan M. Al-Shalabi, Investigation and Analysis of ANN Parameters, European Journal of Scientific Research ,ISSN 1450-216X / 1450-202X Vol.121 No.2, 2014, pp.217-225.
5. Dr. Ghazi. M. Qaryouti, Dr.Saleh Khawatreh, Prof. Ziad A.A. Alqadi, Prof. Mohammed Abu Zalata, Optimal Color Image Recognition System (OCIRS), International Journal of Advanced Computer Science and Technology. ISSN 2249-3123 Volume 7, Number 1 (2017), pp. 91-99.
6. Akram A. Moustafa, Ziad A. Alqadi, Color Image Reconstruction Using a New R'G'I Model, Journal of Computer Science 5 (4): 250-254, 2009.
7. Mohammed Ashraf Al Zudool, Saleh Khawatreh, Ziad A. Alqadi, Efficient Methods used to Extract Color Image Features, IJCSMC, Vol. 6, Issue. 12, pg.7–14, December 2017.
8. Dr. Ziad A.AlQadi, Dr. Hussein M.Elsayyed, Window Averaging Method to Create a Feature Victor for RGB Color Image, IJCSMC, Vol. 6, Issue. 2, pg. 60 –66, February 2017.
9. Hong X, Zhao G, Pietikainen M, Chen X (2014) Combining LBP difference and feature correlation for texture description. IEEE Transactions on Image Processing 23(6):2557–2668.
10. Gupta R, Patil H, Mittal A (2010) Robust order-based methods for feature description. In: 2010 IEEE conference on computer vision and pattern recognition (CVPR).
11. Hanane. Rami, Mohammed. Hamri, Lhoucine. Masmoudi, Objects Tracking in Images Sequence Using Center-Symmetric Local Binary Pattern (CS-LBP), International Journal of Computer Applications Technology and Research Volume 2–Issue 5, 504 -508, 2013.
12. Jamil S. AL-Azzeh: Improved testability method for mesh-connected VLSI multiprocessors: Jordanian Journal of Computers and Information Technology August 2018.
13. Jamil AL-Azzeh, Bilal Zahran, Ziad Alqadi, Belal Ayyoub and Mazen Abu-Zaher: A Novel Zero-Error Method to Create a Secret Tag for an Image; Journal of Theoretical and Applied Information Technology 15th July 2018.
14. Jamil AL-Azzeh, Bilal Zahran and Ziad Alqadi: Salt and Pepper Noise: Effects and Removal; International Journal on Informatics Visualization July 2018.
15. Jamil AL-Azzeh, Oleksandr Kovalenko , Oleksii Smirnov Anna Kovalenko , Serhii Smirnov : Qualitative risk analysis of software development ; Asian Journal of Information Technology July 2018.
16. Bilal Zahran, Jamil Al-Azzeh ,Ziad Alqadi, Mohd-Ashraf Al Zoghoul : A Modified Lbp Method To Extract Features From Color Images : Journal of Theoretical and Applied Information Technology May 2018.
17. Jamil AL-Azzeh, Information Technologies for Supporting Administrative Activities of Large Organizations; DESIDOC Journal of Library & Information Technology, Vol. 38, No. 3, May 2018.

18. Jamil S. AL-Azzeh: A Distributed Multiplexed Mutual Inter-Unit in-Operation Test Method for Mesh-Connected VLSI Multiprocessors; Jordan Journal of Electrical Engineering; 2017 Volume 10, Number 5.
19. Jamil S. AL-Azzeh: Fault-Tolerant Routing in Mesh-Connected Multicomputer based on Majority-Operator-Produced Transfer Direction Identifiers; Jordan Journal of Electrical Engineering Volume 3, Number 2, April 2017.
20. Jamil S. AL-Azzeh, Mazin Al Hadidi, R. Odarchenko,S. Gnatyuk, Z. Shevchuk :Analysis of Self-Similar Traffic Models in Computer Networks; International Review on Modelling and Simulations; October 2017 Volume 10, Number 5.
21. Jamil Al Azzeh, Ziad Alqadi Qazem, M. Jabber: Statistical Analysis of Methods Used to Enhanced Color Image Histogram; XX International Scientific and Technical Conference; Russia May 24-26, 2017.
22. Mazen Abuzaher, Jamil AL-Azzeh: JPEG Based Compression Algorithm; International Journal of Engineering and Applied Sciences Volume 4, Number 4, 2017
23. Mazin al hadidi, Jamil s. Al-azzeh, oleg p. Tkalich,roman s. Odarchenko,sergiy o. Gnatyuk and yulia ye. Khokhlachova2: Zigbee, Bluetooth and Wi-Fi Complex Wireless Networks Performance Increasing; International Journal On Communications Antenna And Propagation, vol 7 No 1 February 2017.
24. Jamil Al Azzeh, Daniel Monday Afodigbokwu, Denis Olegovich Bobyntsev, Igor Valerievich Zotov: Implementing Built-In Test in Analog and Mixed-Signal Embedded-Core-Based System-On-Chips; Asian Journal of Information Technology, Medwell Journals, 2016. (SJR indicator = 0.11).
25. Jamil Al Azzeh, Hussein Alhatamleh, Ziad A. Alqadi, Mohammad Khalil Abuzalata : Creating a Color Map to be used to Convert a Gray Image to Color Image; International Journal of Computer Applications (0975 – 8887).Volume 153 – No2, November 2016.
26. Jamil Al-Azzeh: Analysis of Second Order Differential Equation Coefficients Effects on PID Parameters International Journal on Numerical and Analytical Methods in Engineering (IRENA) Vol 4, No 2 2016.
27. Dmitriy Skopin and Jamil Al-Azzeh; Automated Demodulation of Amplitude Modulated Multichannel Signals with Unknown Parameters Using 3D Spectrum Representation Research Journal of Applied Sciences, Engineering and Technology, Maxwell Scientific Publication June 05, 2016; (SJR indicator = 0.15).
28. Mazin Al Hadidi, Jamil S. Al-Azzeh, R. Odarchenko, Sergiy Gnatyu,k and A. A bakumova Adaptive Regulation of Radiated Power Radio Transmitting Devices in Modern Cellular Network Depending on Climatic Conditions. Contemporary Engineering Sciences, Vol. 9, 2016, no. 10, 473 - (impact factor= 0.193) 2016. 485
29. Mazin Al Hadidi, Jamil S. Al-Azzeh, B. Akhmetov, O. Korchenko,S. Kazmirchuk, M. Zhekambayeva: Methods of Risk Assessment for Information Security Management International Review on Computers and Software (I.RE.CO.S.), Vol. 11, N. 2 ISSN 1828-6003 February 2016.
30. Jamil Al Azzeh, Bidirectional Virtual Bit-slice Synchronizer: A Scalable Solution for Hardware-level Barrier Synchronization. Research Journal of Applied Sciences, Engineering and Technology, 11(8): 902-909. Maxwell Scientific Publication Corp November 2015.
31. Jamil Al Azzeh, Michael E. Leonov. Dniitriy E. Skopm. Evgeny A. Titenko, Isor V Zotov; The Organization of Built-in Hardware-Level Mutual Self-Test in Mesh-Connected VLSI Multiprocessors; International Journal on Information Technology (I.RE.I.T.) Vol. 3, Praise Worthy Prize, March 2015.
32. Jamil Al Azzeh, Dmitriy B. Borzov2, Igor V. Zotov3 and Dmitriy E. Skopin'; an approach to achieving increased fault-tolerance and availability of multiprocessor-based computer systems" ; Australian Journal of Basic and Applied Sciences. Apr. 2014.
33. Jamil Al -Azzeh,S. F. Yatsun, A.A. Cherepanov, I.V. Lupehina4 and V.S. Dichenko; Computer simulation of vibration robot created for the wall movement; Research Journal of Applied Sciences.; 2014 , Issue: 9, Page No.: 597-602 .

34. AL-Azzeh Jamil, Review of Methods of Distributed Barrier Synchronization of Parallel Processes in Matrix VLSI Systems, International Review on Computers and Software (IRECOS), Praise Worthy Prize, Part A, vol. 8, no. 4, pp.42- 46, April 2013 ISSNJS2S-6003
35. Skopin Dmitriy, Al-Azzeh Jamil, Nader Jihad And Abu-Ein Ashraf, Australian Journal Of Basic And Applied Sciences. Dec 2013, Vol. 7 Issue 14, p83-89. 7p. Fastest Color Model For Image Processing Using Embedded Systems.
36. Jamil Al-Azzeh, Mazin Al Hadidi , Using Virtual Network to Solve Freight Company Problems; World Applied Sciences Journal 27 (6): 754-758, 2013.

