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: 7.056

IJCSMC, Vol. 10, Issue. 3, March 2021, pg.55 – 65

Analysis of FFANN Used for Pattern Recognition

Dr. Hatim Ghazi Zaini*; Prof. Ziad AlQadi**
*Taif University, Computer and Information Technology College, KSA
**Albalqa Applied University, Faculty of engineering technology, Jordan
DOI: 10.47760/ijcsmc.2021.v10i03.007

Abstract: FFANN is a powerfull computational model used in various vital applications, these applications require exccelent performane by minimizing the mean square error between the target and the calculating output achieving a high value of recognition ratio. Two methods of creating FFANN model will be introduced and analysized, it will be shown how the programming model of FFANN will increase the recognition ratio and how it is easy to build, train and test this model, this model will be adjustable to suit any input data set and target.

Keywords: FFANN, model architecture, nnstart, progamming model, confusion matrix, ROC, performance.

Introduction

Feedforward artificial neural network [1], [2], [3]is a powerfull computational model used in various fetal applications such as curve fitting(regresion analysiss), pattern recognition(classification) , clustering and and time series[4], [5].

FFANN is a set of fully connected neurons[6]; [7], arranged in layers as shown in figure 1, each neuron acts as a computational cell and performs two main functions [8], [9]as shown in figure 2. The first fuctions is sumation of products of the inputs and the associated weights, the second function is computing the neuron outputs depnding on the activation function (logsig, tansig or linear) selected for the neuron within a specified layer[10], [11].

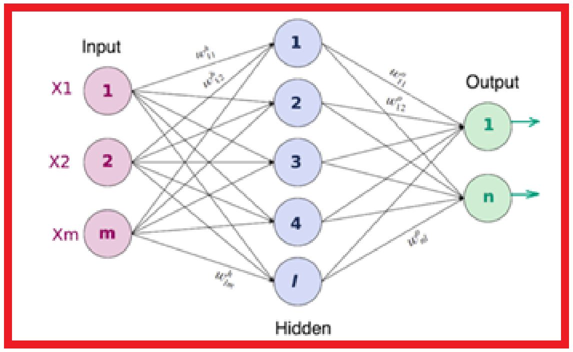


Figure 1: FFANN architecture example

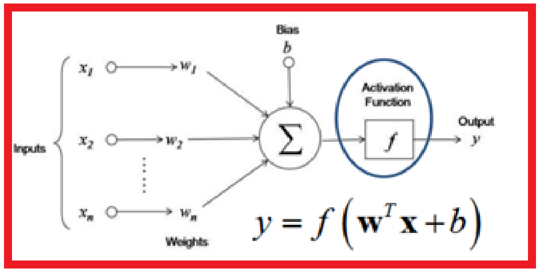


Figure 2: Neuron operations

To use FFANN as a recognition tool we have to follow up the following procedures[12],[13]:

1. Select the input dataset, if required normalize the data.
2. Create FFANN (select FFANN architecture) by defining the number of layers, the number of neurons in each layer, and the activation function for each layer[17], [18].
3. Initialize FFANN.
4. Define some parametors for the net, such as the gaol (the error between the target and the calculated output must equal or closed to zero), the number of training cycles(epochs)[15], [16].

5. Train the net using the inputs and the targets.
6. Chech the error value, if the error is acceptable save the net to be used later as a a recognition tool, else increase the number of training cycle or adjust the net architecture and train it again.

Each training cycle contains two phases as shown if figures 3 and 4. The feedforward phase by calculating the neurons outputs starting from the input layer, and backword phase staring from the ouput layer to adjust the neurons weight[19], [20].

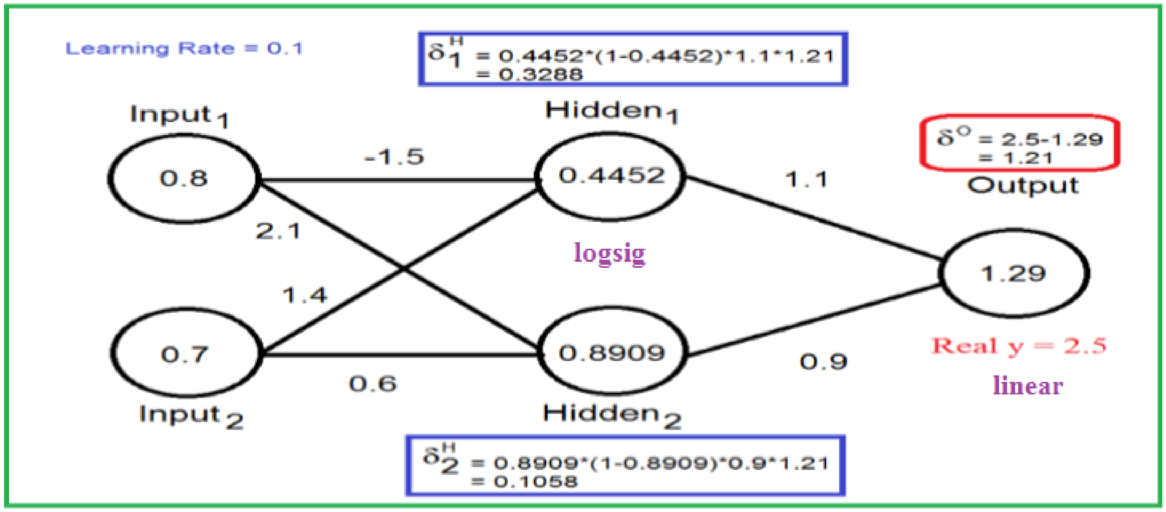


Figure 3: Forward phase

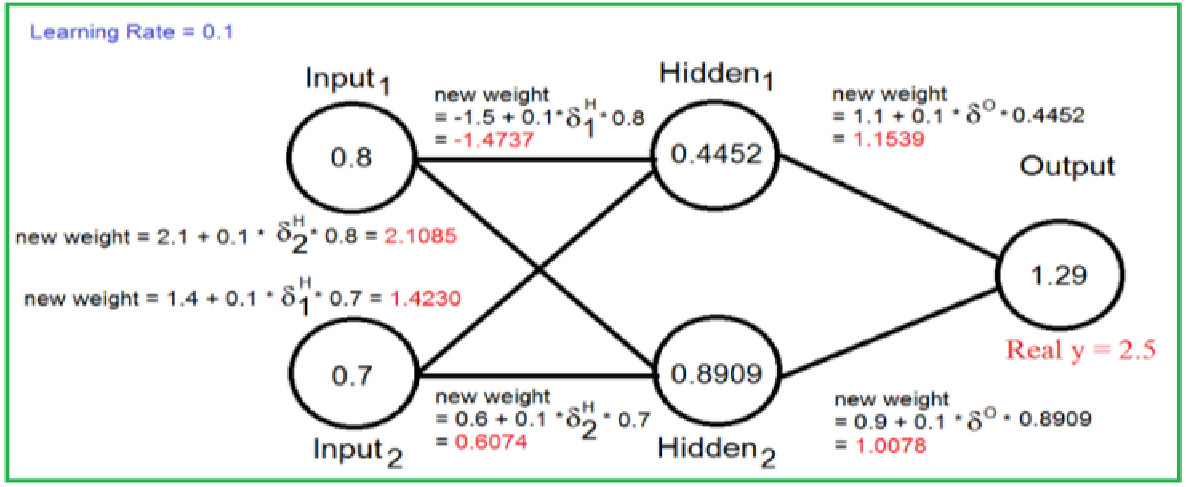


Figure 4: Packword phase

Data set descriptrion

Pattern recognition is the process of training a neural network to assign the correct target classes to a set of input patterns. Once trained the network can be used to classify patterns it has not seen before[13], [14].

This dataset can be used to design a neural network that classifies cancers as either benign or malignant depending on the characteristics of sample biopsies.

The input cancer Inputs data set is a 9x699 matrix defining nine attributes of 699 biopsies.

1. Clump thickness 2. Uniformity of cell size 3. Uniformity of cell shape 4. Marginal Adhesion 5. Single epithelial cell size 6. Bare nuclei 7. Bland chomatin 8. Normal nucleoli 9. Mitoses

The target cancer Targets output is a 2x966 matrix where each column indicates a correct category with a one in either element 1 or element 2. 1. Benign 2. Malignant Figure 5 shows 10 samples from the input and the target data sets.

Input data set	0.2000	0.2000	0.5000	0.5000	0.5000	0.2000	0.3000	1.0000	1.0000	0.4000
	0.1000	0.1000	0.1000	0.4000	0.3000	0.3000	0.5000	0.5000	0.9000	0.1000
	0.1000	0.1000	0.1000	0.6000	0.3000	0.1000	0.7000	0.6000	0.8000	0.1000
	0.1000	0.1000	0.1000	0.8000	0.1000	0.1000	0.8000	1.0000	0.7000	0.1000
	0.2000	0.2000	0.2000	0.4000	0.2000	0.3000	0.8000	0.6000	0.6000	0.2000
	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.9000	1.0000	0.4000	0.1000
	0.2000	0.3000	0.2000	0.8000	0.2000	0.1000	0.7000	0.7000	0.7000	0.3000
	0.1000	0.1000	0.1000	1.0000	0.1000	0.1000	1.0000	0.7000	1.0000	0.1000
	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.7000	1.0000	0.3000	0.1000
Target	**1**	**1**	**1**	**0**	**1**	**1**	**0**	**0**	**0**	**1**
	0	**0**	**0**	**1**	**0**	**0**	**1**	**1**	**1**	**0**

Figure 5: Inputs and targets samples

Using matlab nnstart to create FFANN recognizer

The creation of a classification model involves the following stages (see figure 6):

1. Data preparation (importing, processing, exploration and statistical analysis) , this stage divides the data into two or three parts:
- training data – will be used to build the model
- validation data (in more complex cases) – will be used for evaluation of model quality during its creation
- testing data – will be used to establish the final quality of the model

2. Model creation (using training and optionally validation)
3. Model quality assessment (testing the created model on testing data)
4. Model application and subsequent monitoring (periodical checks if the quality of predictions does not deteriorate over time, for instance due to demographic or market changes)

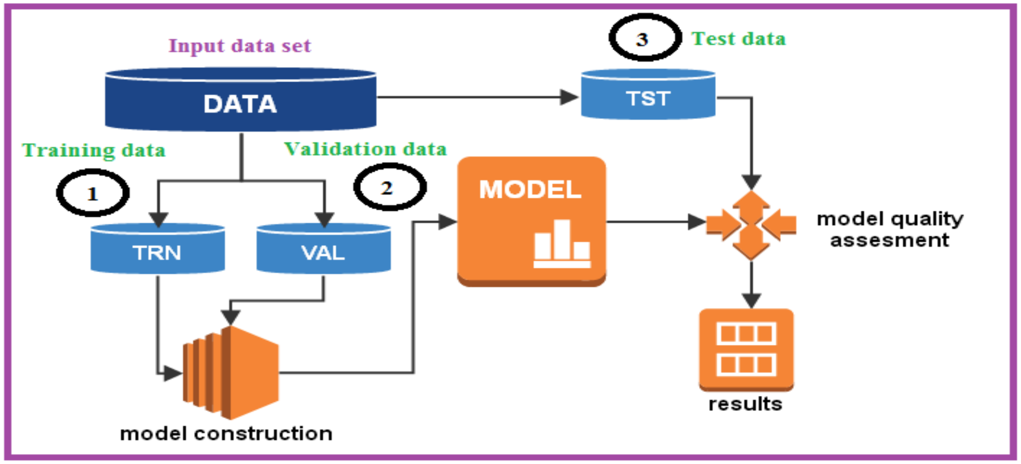


Figure 6: Creating FFANN model using nnstart

Binary classification:

- one class is defined as positive (also known as target class, rare class or minority class)
- other class is defined as negative (also known as normal class)

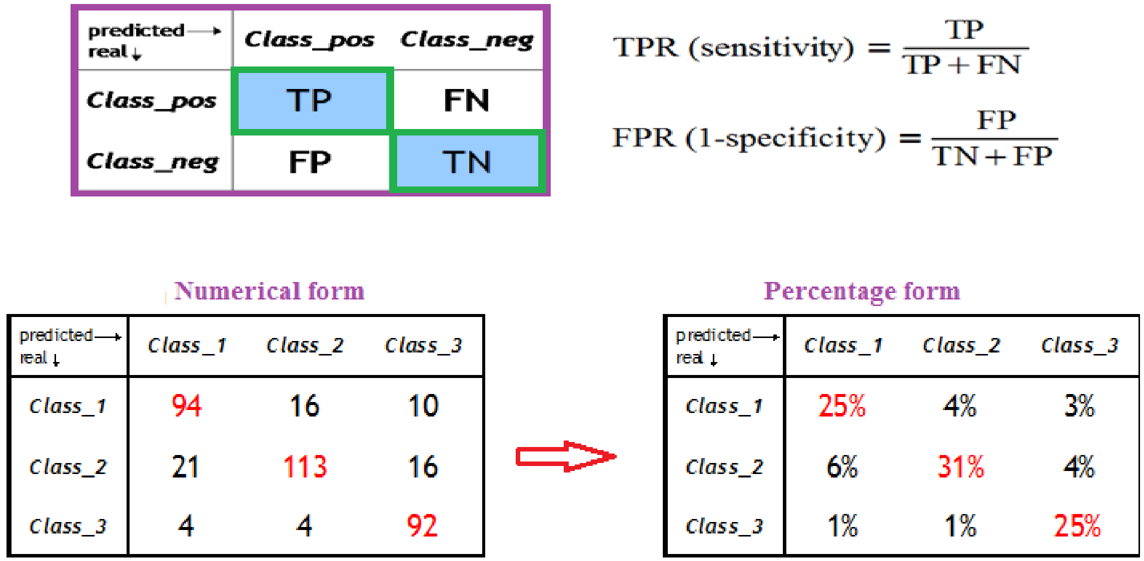
Multiclass classification:

- one class is defined as positive
- other classes combined are defined as negative

Positive class should collect objects which should be identified during modeling: for example, in churn modeling the positive class would consist of resigning customers; in credit scoring projects the positive class consists of customers who defaulted on their debts. (In both cases the negative class consists of the remaining customers).

TP, TN, FP, FN (see figures 7 and 8)

- TP – True Positive – the number of observations correctly assigned to the positive class Example: the model's predictions are correct and resigning customers have been assigned to the class of "disloyal" customers
- TN – True Negative – the number of observations correctly assigned to the negative class Example: the model's predictions are correct and customers who continue using the service have been assigned to the class of "loyal" customers.
- FP – False Positive – the number of observations assigned by the model to the positive class, which in reality belong to the negative class. Example: unfortunately the model is not perfect and made a mistake: some customers, who continue using the service have been assigned to the class of "disloyal" customers.
- FN – False Negative – the number of observations assigned by the model to the negative class, which in reality belong to the positive class. Example: unfortunately the model is not perfect and made a mistake: some churning customers have been assigned to the class of "loyal" customers.



predicted → real ↓	Class_pos	Class_neg
Class_pos	TP	FN
Class_neg	FP	TN

$$\text{TPR (sensitivity)} = \frac{TP}{TP + FN}$$

$$\text{FPR (1-specificity)} = \frac{FP}{TN + FP}$$

Numerical form

predicted → real ↓	Class_1	Class_2	Class_3
Class_1	94	16	10
Class_2	21	113	16
Class_3	4	4	92

Percentage form

predicted → real ↓	Class_1	Class_2	Class_3
Class_1	25%	4%	3%
Class_2	6%	31%	4%
Class_3	1%	1%	25%

Figure 7: Confusion matrix

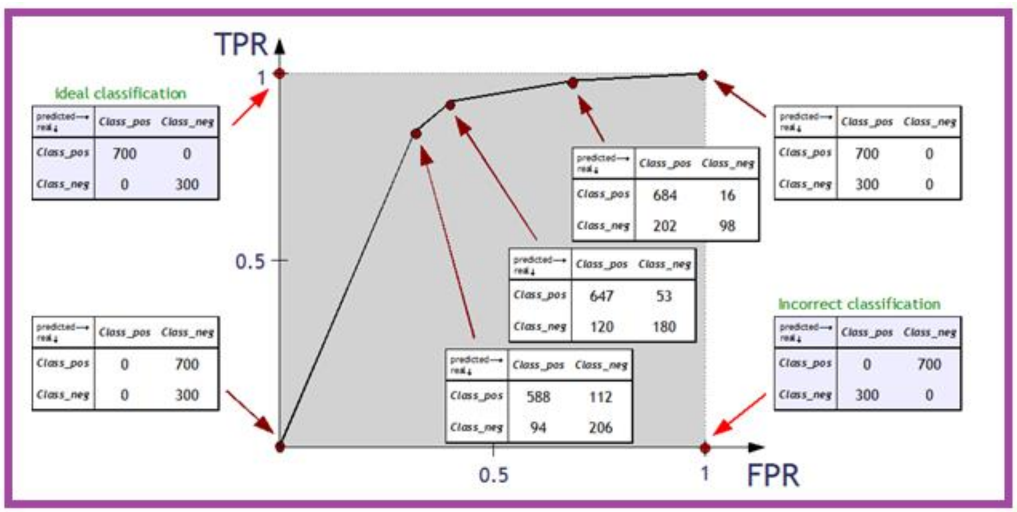


Figure 8: Receiver operation characteristics (ROC)

For a perfect classifier (i.e. every observation has been correctly classified) we would have:
FP = 0
FN = 0
TP = number of all observations from the positive class.
TN = number of all observations from the positive class.
Pos = TP + FN – number of all observations which in reality belong to the positive class.
Neg = FP + TN – number of all observations which in reality belong to the negative class.
Using the above mentions data set FFANN model shown in figure 9 was created, trained, validated and tested, the input data set was divided into three parts, and the best results were achieved using 70 % of the input data as a training set, 15% of input data as a validation set and 15% of the input data as a testing set.

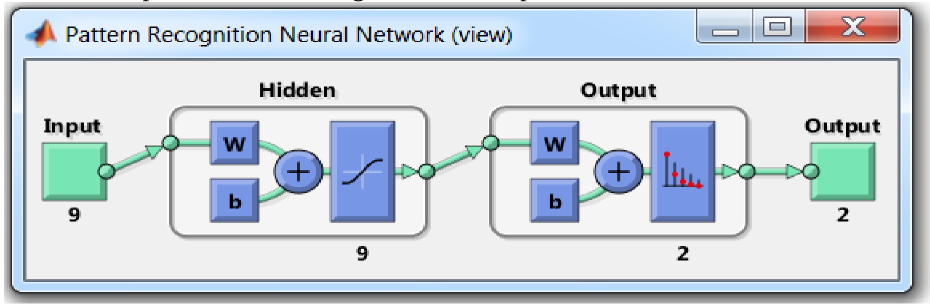


Figure 9: FFANN model architecture

The best recognition ratio obtained was equal 96.5%, this is shown in figures 10, and 11.

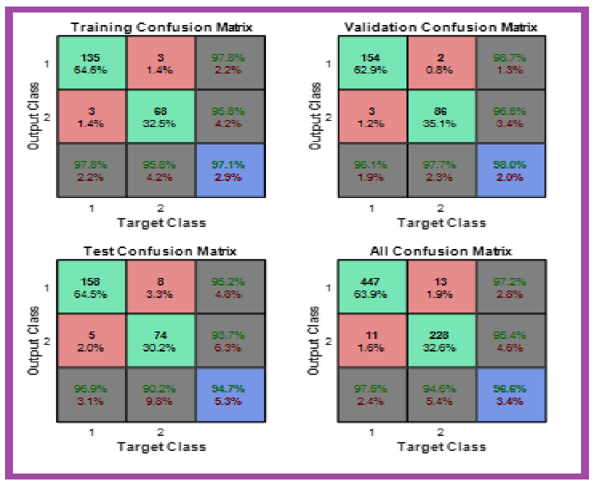


Figure 10: Best confusion matrix

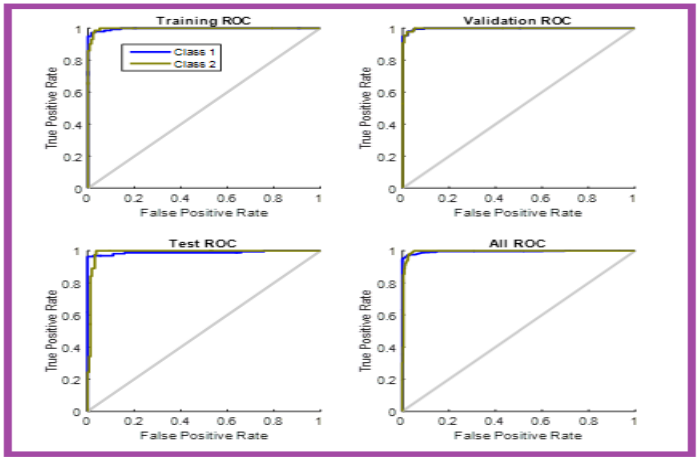


Figure 11: Best ROC

Creating FFANN model using programming method

The following matlab code was used to create and train FFANN with two layers,the obtained FFANN was tested and the error between the targets and the computed output was very closed to zero (MSE = 1.55197e-14)(see figure 12), which means that the model gives a 100% recognition ratio.

```
clear all
load x
load y
net21=newff(minmax(x),[9 2],{'tansig','logsig'});
net21=init(net21);
net21.trainParam.goal=0;
net21.trainParam.epochs=4000;
net21=train(net21,x,y);
y3=sim(net21,x);
```

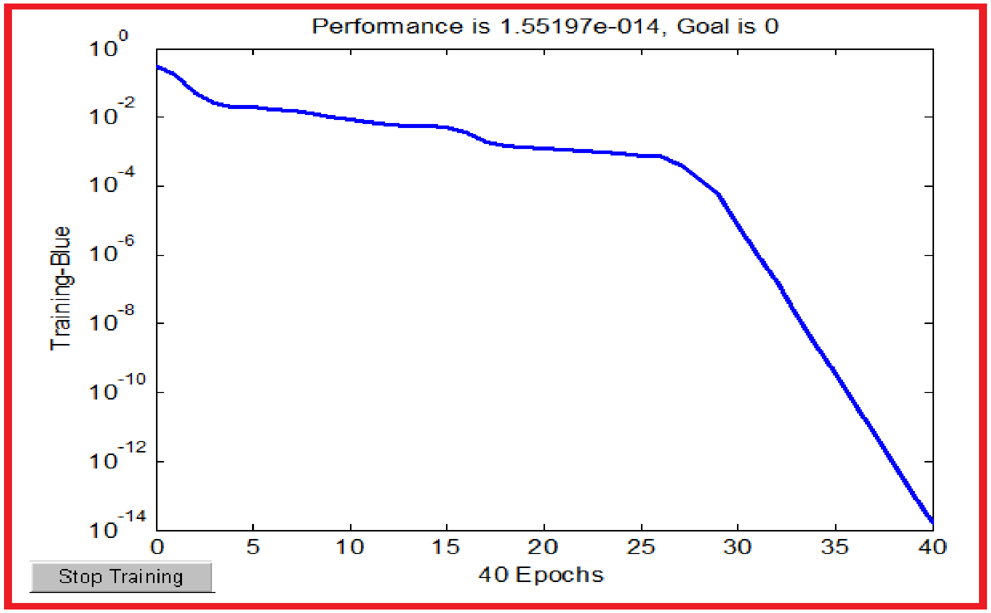


Figure 12: Estimated performance of the programming model

Conclusion

Different ways of creating FFANN model which can be used as recognition tool were introduced, nnstart can be easily used to create a model with good recognition ratio by selecting a suitable model architecture and a suitbale division of the input data set. The worst case of recognirtion ratio is not less than 94% and the best case is 97%. To enhance the recognition ratio and to get a zero error between the targats and the calculated output it is easily to se a programed model, this model is very simple and it is very easy to adjust it to suit any input data set and targets.

References

[1]. Majed O Al-Dwairi, Ziad A Alqadi, Amjad A Abujazar, Rushdi Abu Zneit, Optimized true-color image processing, World Applied Sciences Journal, vol. 8, issue 10, pp. 1175-1182, 2010.

[2]. AlQaisi Aws, AlTarawneh Mokhled, A Alqadi Ziad, A Sharadqah Ahmad, Analysis of Color Image Features Extraction using Texture Methods, TELKOMNIKA, vol. 17, issue 3, 2018.

[3]. Mohammed Ashraf Al Zudool, Saleh Khawatreh, Ziad A. Alqadi, Efficient Methods used to Extract Color Image Features, IJCSMC, vol. 6, issue 12, pp. 7-14, 2017.

[4]. Bilal Zahran Belal Ayyoub, Jihad Nader, Ziad Al-Qadi, Suggested Method to Create Color Image Features Victor, Journal of Engineering and Applied Sciences, vol. 14, issue 1, pp. 2203-2207, 2019.

[5]. Amjad Y Hindi, Majed O Dwairi, Ziad A AlQadi, A Novel Technique for Data Steganography, Engineering, Technology & Applied Science Research, vol. 9, issue 6, pp. 4942-4945, 2019.

[6]. 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, vol. 153, issue 2, pp. 31-34, 2016.

[7]. Rushdi Abu Zneit, Jamil Al-Azzeh, Ziad Alqadi, Belal Ayyoub, Ahmad Sharadqh, Using Color Image as a Stego-Media to Hide Short Secret Messages, International Journal of Computer Science and Mobile Computing, vol. 8, issue 6, pp. 106-123, 2019.

[8]. Jihad Nadir, Ziad Alqadi, Ashraf Abu Ein, Classification of Matrix Multiplication Methods Used to Encrypt-decrypt Color Image, International Journal of Computer and Information Technology, vol. 5, issue 5, pp. 459-464, 2016.

[9]. A.Z Alqadi, Mohammed K Abu Zalata, Ghazi M Qaryouti, Comparative analysis of color image steganography, International Journal of Computer Science and Mobile Computing, vol. 5, issue 11, pp. 37-43, 2016.

[10]. Ziad AlQadi, M Elsayyed Hussein, Window Averaging Method to Create a Feature Victor for RGB Color Image, International Journal of Computer Science and Mobile Computing, vol. 6, issue 2, pp. 60-66, 2017.

[11]. Dr Rushdi S Abu Zneit, Dr Ziad AlQadi, Dr Mohammad Abu Zalata, A Methodology to Create a Fingerprint for RGB Color Image, IJCSMC, vol. 6, issue 1, pp. 205-212, 2017.

[12]. Ahmad Sharadqh Naseem Asad, Ismail Shayeb, Qazem Jaber, Belal Ayyoub, Ziad Alqadi, Creating a Stable and Fixed Features Array for Digital Color Image, IJCSMC, vol. 8, issue 8, pp. 50-56, 2019.

[13]. Ahmad Sharadqh Jamil Al-Azzeh, Rashad Rasras, Ziad Alqadi, Belal Ayyoub, Adaptation of matlab K-means clustering function to create Color Image Features, International Journal of Research in Advanced Engineering and Technology, vol. 5, issue 2, pp. 10-18, 2019.

[14]. ZIAD ALQADI, A MODIFIED LBP METHOD TO EXTRACT FEATURES FROM COLOR IMAGES, Journal of Theoretical and Applied Information Technology, vol. 96, issue 10, pp. 3014-3024, 2018.

[15]. Aws Al-Qaisi, Saleh A Khawatreh, Ahmad A Sharadqah, Ziad A Alqadi, Wave file features extraction using reduced LBP, International Journal of Electrical and Computer Engineering, vol. 8, issue 5, 2018.

[16]. Ahmad Sharadqh Naseem Asad, Ismail Shayeb, Qazem Jaber, Belal Ayyoub, Ziad Alqadi, Creating a Stable and Fixed Features Array for Digital Color Image, IJCSMC, vol. 8, issue 8, pp. 50-56, 2019.

[17]. Jamil Al-Azzeh, Ziad Alqadi, Mohammed Abuzalata, Performance Analysis of Artificial Neural Networks used for Color Image Recognition and Retrieving, international Journal of Computer Science and Mobile computing, vol. 8, issue 2, pp. 20-33, 2019.

[18]. 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, vol. 121, issue 2, pp. 217-225, 2014.

[19]. Prof. Mohammed Abu Zalata Dr. Ghazi. M. Qaryouti , Dr.Saleh Khawatreh, Prof. Ziad A.A. Alqadi, Optimal Color Image Recognition System (OCIRS), International Journal of Advanced Computer Science and Technology., vol. 7, issue 1, pp. 91-99, 2017.

[20]. Abdullah Al-Hasanat, Haitham Alasha'ary, Khaled Matrouk, Ziad Al-Qadi, Hasan Al-Shalabi, Experimental Investigation of Training Algorithms used in Back propagation Artificial Neural Networks to Apply Curve Fitting, European Journal of Scientific Research, vol. 121, issue 4, pp. 328-335, 2014.

