Identification of Seamless Connection in Merged Images using Evolutionary Artificial Neural Network (EANN)

L. Ali[1], A. Sarwar[2], A. Jan[3], M. I. Khattak[4], M. Shafi[5], Najvia[6]

[1,2,3,4,5,6]*Electrical Engineering Department, UET Peshawar, Pakistan*
[4]M.I.Khattak@uetpeshawar.edu.pk

Abstract-Recently various software have been invented in order to make the job of image manipulation more comfortable and effective. Using software images are merged in such a way that ocular review cannot differentiate the resulting forgeries from authentic images. This paper proposes a model for the identification of such merged images. Proposed methodology for identification is the connection of two stagesfeature extraction using principle component analysis and the use of classifier based on the evolutionary artificial neural network (EANN). The behavioral aspects of EANN phenotype was evaluated under changing parameters. The results obtained after experimentation were promising.

I. NOMENCLATURE

ANN	Artificial Neural Network
EANN	Evolutionary Artificial Neural Network
BNN	Biological Neural Network
PCA	Principal Component Analysis
GA	Genetic Algorithm

II. INTRODUCTION

In this era of modern technology image manipulation is one of the most common practices. Using various sophisticated software like Photoshop images are fused in such a way that an individual is remarkably unable to identify whether image is merged or not. Image forging is carried out commonly for deceitfulness.

Certain factors in images are added with the best intensions to make a merged image more believable and more likely to be imprinted on individual memories [i]. Although image merging has a lot of social, political and religious issues associated with it [ii]. A great deal of manipulated images are used for different purposes, black mailing is one of the major problem of the day especially on social media sites like Facebook and Tweeter [iii-iv]. Various countries like France and Britain have taken action against the prevalence of forged images; they claim that these unreal images are harmful to individual's psyche, that is why majority of countries have imposed restriction upon uploading of such images on social media sites [ii]. Besides these restrictions they are unable to control the spreading of manipulated images worldwide. There should be some efficient technique to classify manipulated images before it creates a fuss in social media and effect individuals psyche. This paper proposes a model for the classification of forged images.

Literature review shows that high performance image classification can be exercised with the help of feature extracting schemes and by using machine learning algorithms [v-vi]. The main problem in developing merged image identification system is to select relevant feature extracting scheme and classification system that can achieve any desired goals. Feature extraction is essential in identification process because images are too complex to be processed by any classification system [vii]. Features are extracted from images using various schemes including Fast Fourier Transform (FFT), Discrete Wavelet Transform (DWT) and Principle component analysis (PCA) to name a few. However, FFT is a good technique for frequency analysis while DWT is desirable to reduce dimensionality of data and for the data analysis in both frequency and spatial domain [vii]. Ideally the set of features used in classification should be independent of one another. Individual can use PCA which is an effective technique [viii], to obtain independent features and to eliminate redundant information [ix]. Feature measurements of images are passed to classifier that evaluate the features of images and affiliate to which category they belong e.g. in the case of email, classifiers colligate whether email is spam or not. There are several classifiers such as decision trees, neural networks, naïve Bayesian classifier for decision support system [x]. For highly accurate classification most of the classification systems used today are Artificial Neural Network (ANN) [xi-xii], but according to Xin Yao better intelligent system can be created if evolutionary algorithms (EAs) and ANNs are used in combination [xiii] known as Evolutionary artificial neural network EANN.

In this paper the main subject addressed is to identify the forged images using PCA and EANN as feature extracting scheme and classification algorithm respectively. Here EANN is used as classifier whose accuracy of identification strongly depend on the quality of training data as well as the structure and training algorithm chosen [xiv]. The training data chosen for identification purpose are the features extracted from images using PCA while Genetic Algorithm (GA) is used to train the ANN forming EANN. Individual can find Neuro Evolution ANN application in nonlinear control problem, robot controlling cancer detection, spam email image identification, face recognition, image authentication, image segmentation, image compression and several other aspects of image processing [xii], [xv-xxix].

Rest of the paper is organized as follows: Section III provides a brief background of EANN and its operation, followed by the EANN architecture used for merged image identification process. Section IV describes the methodology used to classify merged and original images. It includes the creation of database and feature extraction process in detail. Section V and VI provides a detailed discussion on the training and testing of EANN. Section VII, includes the EANN training and testing results. Finally, a brief summary and conclusion are in Section VIII.

III. Evolutionary Artificial Neural Network (EANN)

EANN is a computational classification system based on the functionality and structure of biological neural network (BNN). According to Xin Yao EANN originates when one of the evolutionary algorithms such as evolution strategies (ES), evolutionary programing (EP) and genetic algorithm (GA) are used to train an artificial neural network [xxx]. In this paper GA, working as genotype, is used to find the optimum connecting weights (genes) for fixed ANN topology known as phenotype. In EANN genotype is a long string created only from weights of ANNsynaptic connections. EANN phenotype comprise of nodes having same functionality as neuron in biological neural network (BNN). Fig. 1 shows the computational node in EANN phenotype, nodes have several inputs and function associated with them; these inputs can be from neighboring nodes or from program inputs and associated function of each node is a sigmoid function. Subsequent subsection put some light on EANN functionality and architecture.

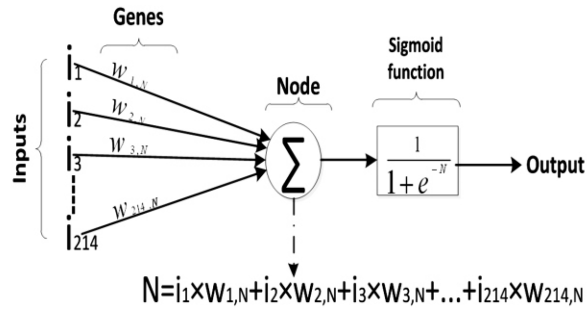


Fig.1.Computational Neuron Architecture

A. *EANN Algorithm Functionality*

For having a good grasp on how EANN operates, the fundamental parts in Fig. 2 which accounts for the EANN operation are described in detail as follow

1) Initialization

In the initialization, all the weights (genes) in the EANN phenotype (architecture) are joined to make a string called genotype. Random numbers are assigned for each gene (weight) between -1 and 1. This generated string (parent) is then used as a member of initial population as shown in Fig. 3.

2) Mutation

Mutation is to alter individual genes to produce a new individual child. Number of mutants are chosen in accordance to evolutionary strategy $(1 + \lambda)$-ES, which states that λ mutants can be generated from a parent. Mutants compete with the parent for fitness; the fittest mutant turn into a parent of the next generation. Figure 4 shows four mutants (offspring's) that are generated from parent genotype of Fig. 3. Mutation results in $(1+\lambda)$ population size. Mutation is expressed as follow
Number of genes to be mutated = Total genes×Mutation rate

3) Selection

Selection determines which of the individual will survive in a generation. Set of examples are provided as input to EANN and output is calculated for all population size. Selection is based on maximum successful matches of EANN output with the individual target values. Selected genotype will be the fittest among all population size and will survive.

The process of mutation selection and fitness evaluation in each generation continue by evolving the genotype until desired behavior of phenotype is obtained.

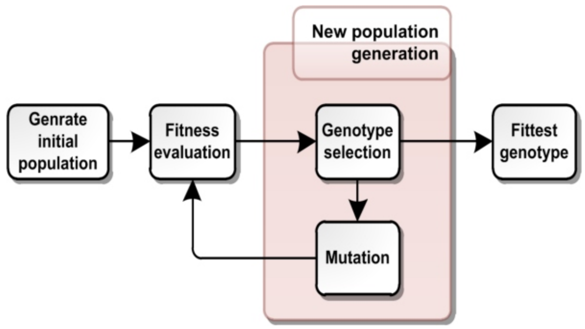


Fig. 2. Procedure of EANN

-0.8015	0.1799	0.0754	-0.4833	-0.7488	-0.7443	-0.3211	...	0.4824

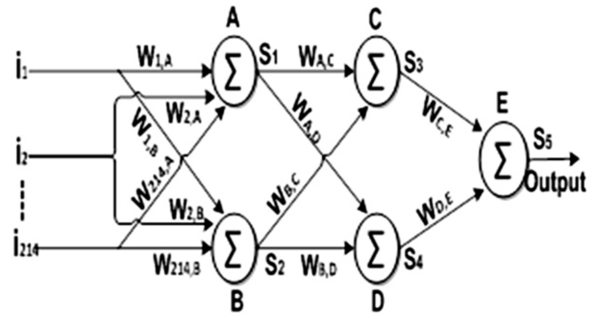
Fig. 3. Parent genotype

-0.5977	0.1799	0.6132	-0.4833	-0.7488	-0.7443	-0.3211	...	0.4824
-0.8015	0.1799	0.0754	-0.7810	-0.7488	-0.7443	-0.3211	...	0.3368
-0.8015	0.1799	0.4746	-0.4833	-0.7488	-0.7443	-0.3211	...	0.5845
-0.8015	0.1799	0.0754	-0.4833	-0.3358	-0.5216	-0.3211	...	0.4824

Fig. 4. Offspring Genotype, colored genes indicate mutation

B. EANN Architecture

EANN phenotype is made up of nodes analogous to neurons in BNN. Nodes in the EANN phenotype works the same as it does in ordinary ANN. Nodes take numerous weighted inputs, sum them and pass the activity 'S' through sigmoid function which produces an output which is always between zero and one. A typical computational node and its function are shown in Fig. 1. EANN phenotype is a collection of such computational nodes. The EANN Architecture chosen for the process of identification is shown in Fig. 5. It has 214 inputs i.e. I_1, i_2, i_3,…, I_214 representing features of single image, weights 'w' in the EANN phenotype representing the genes in its genotype, 2 input nodes viz. 'A' and 'B', 1 hidden layer containing 2 nodes namely 'C' and 'D' and 1 output node 'E'. Output node shows the desired behavior of EANN phenotype i.e.output '1' and '0'will indicate original image and merged image respectively.



$$\mathbf{A = i_1 \times W_{1,A} + i_2 \times W_{2,A} + i_3 \times W_{2,A} + \cdots + i_{214} W_{214,A}}$$
$$\mathbf{S1 = \frac{1}{1+e^{-A}}}$$
$$\mathbf{B = i_1 \times W_{1,B} + i_2 \times W_{2,B} + i_3 \times W_{2,B} + \cdots + i_{214} W_{214,B}}$$
$$\mathbf{S2 = \frac{1}{1+e^{-B}}}$$
$$\mathbf{C = S_1 \times W_{A,C} + S_2 \times W_{B,C}} \qquad \mathbf{S3 = \frac{1}{1+e^{-C}}}$$
$$\mathbf{D = S_1 \times W_{A,D} + S_2 \times W_{B,D}} \qquad \mathbf{S4 = \frac{1}{1+e^{-D}}}$$
$$\mathbf{E = S_3 \times W_{C,E} + S_4 \times W_{D,E}} \qquad \mathbf{S5 = \frac{1}{1+e^{-E}}}$$

Fig. 5. EANN phenotype

IV. METHODOLOGY

The procedure of merged image identification is carried out using machine learning algorithm such as EANN described previously. First, features are extracted from all the images in the database using PCA and are stored in a single text file. Secondly, EANN is trained with training data to learn to distinguish between merged and original images. Finally, the network is tested with "unknown" images to assure how considerably EANN has learned to identify merged and original images. The overall process of merged image identification is shown in Fig. 6.

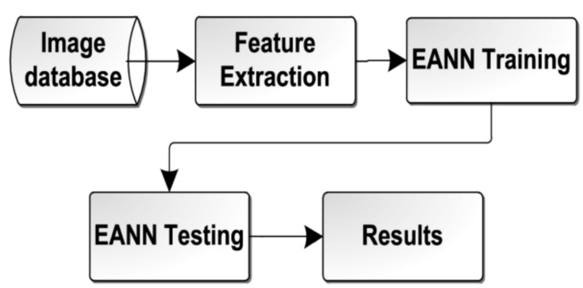


Fig. 6. Methodology of merged image identification using EANN

A. Database Creation

Two databases areused in this paper. One is online database comprise of 100 authentic and 100 forged color images having dimensions of 384×256, taken from internet named as database 1 while self-created database 2 possesses 214 merged and 214 original images having dimensions 480×640 taken from Samsung Galaxy s4 Camera. Photos in database2 are merged with the help of Photoshop editing software. The merging of photos in Photoshop commence as follow.

Open any two images individually you want to merge in Photoshop, make selection using lasso tool or quick select. After selecting lasso tool drag the cursor over the part of pic individual want to merge. Select Refine edge (make changes in dialogue box according to individual need), output what individual have done to new layer with mask. Move the new layer image to background image and adjust its size, apply transformation. Click 'Merge Layers' and save it. All the merged images in database are created in same fashion. The overall process of image merging via Photoshop is shown in Fig. 7.

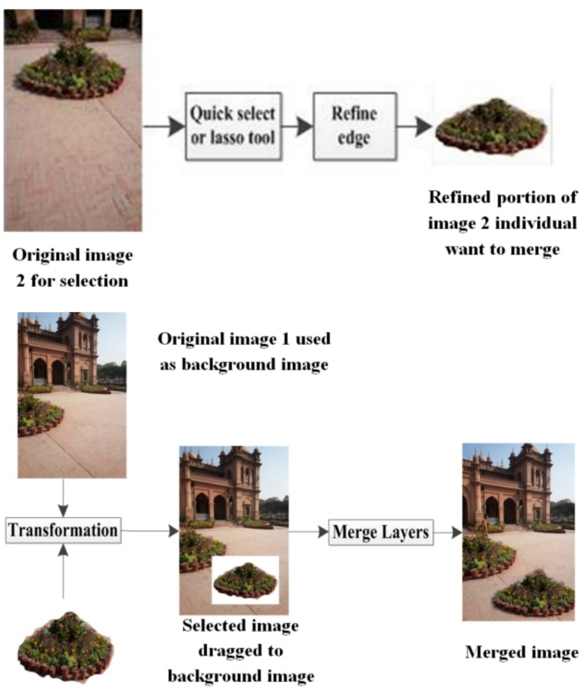


Fig. 7. Merging of photos using Photoshop

B. Feature Extraction

The process of merged images identification commence with the extraction of features from images. As the image is too immense to be processed by the EANN algorithm, PCA is employed to eliminate the redundant information from images.

PCA is a statistical technique that transforms the image to new set of coordinates called the principle component [xxxi]. Principle components are the directions where the data is more spread out i.e. where it has high variance. Fig. 8 and Fig. 9 shows the process of obtaining the principle components.

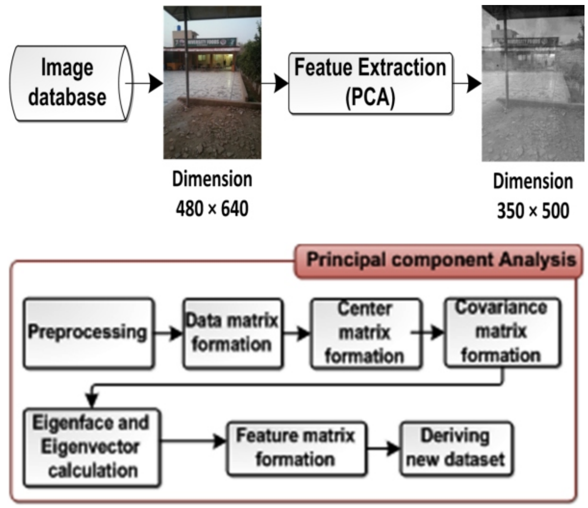


Fig. 8. Block diagram for feature extraction using PCA

-1.07E+08	-3.33E+07	7.39E+07	3.48E+06	-5.44E+06	···	2.44E+07

Fig. 9. Sample final data obtained in the end of feature extracting process

The process begins with preprocessing of all Images in the databases. Preprocessing reshape all the 480×640 and 383×256 dimension images of both database into 1*175000 and 1*98304 dimension vectors. These preprocessed images are arranged in data matrix (D), each column of data matrix D represent reshaped image vectors called observation (N), having M unknown variables, Eq. (1) represents nth observation of data matrix D. For instance the data matrix formed from original images or merged images in database1 have N=214 and M=175000 while N=100 and M=98304 in case of data matrix formed from database2.Aim of data matrix formulation is to reduce M=175000 or M=98304 variables to L=213 or L=99 variables for each observation in the data matrix. Mean of data matrix D is found from Eq. (2) and stored in vector M. After calculating mean, a new matrix CM known as center matrix is calculated using Eq. (3) by subtracting image mean M from the data matrix D. Covariance matrix of the center matrix is then foundusing Eq. (4) by multiplying center matrix and its transpose. Principal component variances i.e. eigenvalues of covariance matrix of center matrix is found using 'princomp' command in MATLAB which returns a 214×214 or 100×100 in case of database 1 and database 2 respectively, of size equal to number of observations. Applying Eq. (5), Eigen faces can be found by the multiplication of center matrix with eigenvalues matrix 'v'. The unnecessary components that have small eigenvalues are discarded and feature vector is formed from remaining eigenvectors as seen by Eq. (5). New data set is formed in terms of these eigenvectors which are used as basis vector given by Eq. (6)

$$D_n=(D_1,D_2 \ldots , D_M)^T \quad (1)$$

Where n=1,2,…,N=428

$$M(m)=\frac{1}{N}\sum_{n=1}^{N} D(m,n) \quad (2)$$

Where m=1,2,…,M

$$CM=D - M*I \quad (3)$$

Where,(I) is identity matrix

$$Cov(CM)=CM.CM^T \quad (4)$$

$$Eigenfaces= CM\times V \quad (5)$$

$$\text{Feature vector} =(eig_1,eig_2 \ldots ,Eig_L) \quad (6)$$

Where L=213 or L=99

$$\text{Final dataset}=\textit{feture vector}^T\times CM \quad (7)$$

Final data set has dimension of 213*428and 200 * 99 data matrix in case of database 1 and database 2 respectively, implying that the value of M for all the 428 or 200 observation has been reduced from 175000 or 98304 variables to just 213 or 99 variables, these 213 or 99 variables in single observation represents the features of an image. Finally, the final data sets are stored in text files, before storing it, few changes are made to final data matrixes, so that the text files are compatible with inputs of EANN and these changes are described in the subsequent paragraph.

For EANN compatible input preparation, transpose of final data matrix is necessitated which results in 428×213 or 200×99data matrix. Each row of resulting matrixes depicts features of single image. Tags known as target values are inserted at 214thand 100thcolumn for image classification, tags '0' and '1' indicates merged and original image respectively. This new matrix of size 428× 214and 200× 100is stored in a text file. The text file is divided into two dataset known as training and testing data sets respectively. Text file prepared from database 1 contain 50% of the data for training and 50% for testing while text file obtained from database 2 contain 90% of data for training and 10% of data for testing. These data sets will be used in the training and testing of EANN. Figure 10 shows the sample input text file obtained from data base provided as input to EANN algorithm.

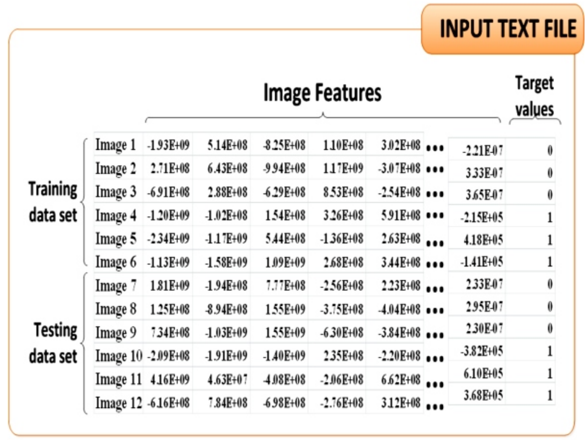


Image 1	-1.93E+09	5.14E+08	-8.25E+08	1.10E+08	3.02E+08	•••	-2.21E07	0
Image 2	2.71E+08	6.43E+08	-9.94E+08	1.17E+09	-3.07E+08	•••	3.33E07	0
Image 3	-6.91E+08	2.88E+08	-6.29E+08	8.53E+08	-2.54E+08	•••	3.65E07	0
Image 4	-1.20E+09	-1.02E+08	1.54E+08	3.26E+08	5.91E+08	•••	-2.15E+05	1
Image 5	-2.34E+09	-1.17E+09	5.44E+08	-1.36E+08	2.63E+08	•••	4.18E+05	1
Image 6	-1.13E+09	-1.58E+09	1.09E+09	2.68E+08	3.44E+08	•••	-1.41E+05	1
Image 7	1.81E+09	-1.94E+08	7.77E+08	-2.56E+08	2.23E+08	•••	2.33E07	0
Image 8	1.25E+08	-8.94E+08	1.55E+09	-3.75E+08	-4.04E+08	•••	2.95E07	0
Image 9	7.34E+08	-1.03E+09	1.55E+09	-6.30E+08	-3.84E+08	•••	2.30E07	0
Image 10	-2.09E+08	-1.91E+09	-1.40E+09	2.35E+08	-2.20E+08	•••	-3.82E+05	1
Image 11	4.16E+09	4.63E+07	-4.08E+08	-2.06E+08	6.62E+08	•••	6.10E+05	1
Image 12	-6.16E+08	7.84E+08	-6.98E+08	-2.76E+08	3.12E+08	•••	3.68E+05	1

Fig. 10. Sample input text file obtained from database1 provided as input to EANN

V. EANN TRAINING

EANN is trained on training data set created previously to learn to distinguish between merged and original image. For training EANN text file similar to that shown in Fig. 8, is read by the MATLAB EANN training program. Each line in text file can be considered as string (chromosome). To make this string binary, the data has been normalized between 0 and 1.A parent genotype similar to a genotype shown in Fig. 3 is constructed in which genes are assigned random numbers between -1 and 1. Parent genotype is mutated with 10% mutation rate, mutation results in 4 offspring's analogous to Figure4, thus a population size of 5 (1+4) i.e. having 1 parent and 4 offspring's has been created. Initially EANN phenotype is constructed using the parent genotype. During the first iteration training dataset is given to EANN phenotype and outputs are calculated for individual imagesin training data set and compared with their respective target values. During next iteration one of the offspring genotype is used to construct EANN phenotype and trained with training data. The process is repeated for all the remaining offspring genotypes. Five count for five genotypes has been initialized, that count successful match of outputs with individual target values. At the termination of first generation all the five counts are compared with one another, the genotype having maximum count will survive and turn into a parent of the next generation. The max count indicates the fittest genotype among the initial population. The process of mutation and selection continue for 1 million generations. The genotype which survives after 1 million generation will be used in testing the behavior of EANN phenotype to examine how substantially EANN has learned to distinguish between merged and original image.

VI. EANN TESTING

The purpose of EANN training was to find compatible connecting weights known as the fittest genotype which survived for 1 million generation. The fittest genotype obtained from the training EANN was used to construct fixed EANN phenotype shown in Figure 5. EANN that has been trained with training data set is now tested with testing data set to figure out how well the EANN has learned. Outputs for the entire individuals in the testing data set are calculated and compared with their respective target values; hence performance of system is evaluated. Accompanying section include the results obtained from training and testing of EANN phenotype with training and testing data respectively.

VII. EANN TRAINING AND TESTING RESULTS

EANN phenotype was trained and tested with different characteristics of phenotype i.e. EANN training and testing algorithm was simulated number of times with different seed values, increasing network size but with fixed mutation rate and same training and testing data set. During each run behavior of phenotype was checked and efficiency was calculated, the results of the experiment are tabulated in Table I.

In the end of experiment 1 it was observed that keeping the network size "2×2" and seed value "0" the efficiency of EANN phenotype was reached up to 82 % but does not exceed this limit while in experiment 2 the accuracy of identification reached up to 100% on keeping the network size "2×2" and seed value 1.

TABLE I
EXPERIMENTAL RESULTS FOR TWO-FOLD TRAINING AND TESTING

Seed Value	Network Size	Training result	Training result efficiency	Testing result	Testing result efficiency
0	2 x 2	184	85.98%	176	**82.24%**
1	2 x 2	150	70.09%	153	71.49%
2	2 x 2	187	87.38%	172	80.37%
0	3 x 3	182	85.04%	174	81.30%
1	3 x 3	178	83.17%	146	68.22%
2	3 x 3	162	75.70%	125	58.41%
0	4 x 4	168	78.50%	152	71.02%
1	4 x 4	141	65.88%	147	68.69%
2	4 x 4	156	72.89%	141	65.88%

TABLE II
EXPERIMENTAL RESULTS FOR TEN-FOLD TRAINING AND TESTING

Seed Value	Network Size	Training result	Training result efficiency	Testing result	Testing result efficiency
0	2 x 2	161	89.4%	18	90%
1	2 x 2	166	92.2%	20	**100%**
2	2 x 2	143	79.4%	15	75%
0	3 x 3	143	79.4%	14	70%
1	3 x 3	161	89.4%	18	90%
2	3 x 3	151	83.8%	17	85%
0	4 x 4	159	88.3%	18	90%
1	4 x 4	164	91.1%	18	90%
2	4 x 4	138	76.6%	15	75%

VIII. CONCLUSION

In this paper, original and merged images are identified with the help of Principle Component Analysis and EANN, which are used for feature extraction and machine learning approach respectively. Since, the images are too immense to be given as an input to EANN, so for dimensionality reduction the feature vectors of images are used. The results obtained from EANN for image identification are promising. Current EANN is being trained and tested with a target of 214 or 100 images. Best results are obtained by keeping the number of nodes in each layer small. Experimentation showed that identification accuracy has reached up to 100% by employing features extracted using PCA technique.

REFERENCES

[i] R. Eveleth, "How fake images change our memory and behaviour," 2014.Available: http://www.bbc.com/future/story/20121213-fake-pictures-make-real-memories

[ii] H. Magee, "Ethical Inquiry: August 2012 | Brandeis University," *Ethical Inquiry,* 2014.Available: http://www.brandeis.edu/ethics/ethicalinquiry/2012/August.html

[iii] M. Yaseen. (2014, September, 4). *Cyber Crimes| Dunya Blog*. Available: http://blogs.dunyanews.tv/?p=6651

[iv] J. Shahid, "Cyberstalking: New challenges - Pakistan - DAWN.COM," 2014. Available:http://www.dawn.com/news/1078417

[v] K. NATARAJ and N. PATNEKAR, "Neural Networks for 1mage Analysis and Processing in Measurements, 1nstrumentation and Related 1ndustrial Applications, "*Neural Networks for*

Instrumentation, Measurement and Related Industrial Applications, vol. 185, p. 145, 2003

[vi] S. Shah and N. Riaz, "Analytical Study of Face Recognition Techniques," *International Journal of Signal Processing, Image Processing & Pattern Recognition,* vol. 6, 2013

[vii] B. Temel, N. Kilic, and B. Ozgultekin, "Separation of original paintings of Matisse and his fakes using wavelet and artificial neural networks," *IU-Journal of Electrical & Electronics Engineering,* vol. 9, pp. 791-796, 2012

[viii] I. S. Bajwa, M. S. Naweed, M. N. Asif, and S. I. Hyder, "Feature based image classification by using principal component analysis," *ICGST-GVIP Journal, ISSN,* 2009

[ix] Y. Zheng and E. A. Essock, "Novel feature extraction method-Wavelet-Fourier analysis and its application to glaucoma classification," in *Proceedings of 7th Joint Conference on Information Sciences*, 2003, pp. 672-675.

[x] S. Kotsiantis and P. Pintelas, "A hybrid Decision Support Tool," in *Proceedings of 6th International Conference on Enterprise Information Systems (ICEIS-2004), Porto-Portugal*, 2004, pp. 14-17.

[xi] D. Michie, D. J. Spiegelhalter, and C. C. Taylor, "Machine learning, neural and statistical classification," 1994

[xii] H. A. Montes and J. L. Wyatt, "Cartesian Genetic Programming for Image Processing Tasks," in *Neural Networks and Computational Intelligence*, 2003, pp. 185-190.

[xiii] X. Yao, "Evolving artificial neural networks," *Proceedings of the IEEE,* vol. 87, pp. 1423-1447, 1999

[xiv] D. Floreano, P. Dürr, and C. Mattiussi, "Neuroevolution: from architectures to learning," *Evolutionary Intelligence,* vol. 1, pp. 47-62, 2008

[xv] M. M. Khan, G. M. Khan, and J. F. Miller, "Evolution of Optimal ANNs for Non-Linear Control Problems using Cartesian Genetic Programming," in *IC-AI*, 2010, pp. 339-346.

[xvi] M. M. Khan, G. M. Khan, and J. F. Miller, "Efficient representation of recurrent neural networks for markovian/non-markovian non-linear control problems," in *Intelligent Systems Design and Applications (ISDA), 2010 10th International Conference on*, 2010, pp. 615-620.

[xvii] F. J. Gomez and R. Miikkulainen, "Solving non-Markovian control tasks with neuroevolution," in *IJCAI*, 1999, pp. 1356-1361.

[xviii] S. Harding and J. F. Miller, "Evolution of robot controller using cartesian genetic programming, " in *Genetic programming*, ed: Springer, 2005, pp. 62-73.

[xix] J. Khan, J. S. Wei, M. Ringner, L. H. Saal, M. Ladanyi, F. Westermann*, et al.*, "Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks," *Nature medicine,* vol. 7, pp. 673-679, 2001

[xx] H. A. Abbass, "An evolutionary artificial neural networks approach for breast cancer diagnosis," *Artificial Intelligence in Medicine,* vol. 25, pp. 265-281, 2002

[xxi] A. M. Ahmad, G. M. Khan, S. A. Mahmud, and J. F. Miller, "Breast cancer detection using cartesian genetic programming evolved artificial neural networks," in *Proceedings of the fourteenth international conference on Genetic and evolutionary computation conference*, 2012, pp. 1031-1038.

[xxii] D. B. Fogel, E. C. Wasson III, and E. M. Boughton, "Evolving neural networks for detecting breast cancer," *Cancer letters,* vol. 96, pp. 49-53, 1995

[xxiii] J. R. Bowling, P. Hope, and K. J. Liszka, "Spam Image Identification Using an Artificial Neural Network," *The University of Akron Akron, Ohio,* pp. 44325-4003, 2008

[xxiv] H. A. Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 20, pp. 23-38, 1998

[xxv] T. Bhattacharya, S. Hore, and S. Bhadra Chaudhuri, "An Image Authentication Technique by Handwritten Signature Verification using DWT and ANN," *International Journal of Computer Applications,* vol. 47, 2012

[xxvi] W. E. Reddick, J. O. Glass, E. N. Cook, T. D. Elkin, and R. J. Deaton, "Automated segmentation and classification of multispectral magnetic resonance images of brain using artificial neural networks," *Medical Imaging, IEEE Transactions on,* vol. 16, pp. 911-918, 1997

[xxvii] G. Kuntimad and H. S. Ranganath, "Perfect image segmentation using pulse coupled neural networks," *Neural Networks, IEEE Transactions on,* vol. 10, pp. 591-598, 1999

[xxviii] S. Costa and S. Fiori, "Image compression using principal component neural networks," *Image and vision computing,* vol. 19, pp. 649-668, 2001

[xxix] S. Harding, J. Leitner, and J. Schmidhuber, "Cartesian genetic programming for image processing," in *Genetic Programming Theory and Practice X*, ed: Springer, 2013, pp. 31-44.

[xxx] X. Yao, "Evolutionary artificial neural networks," *International journal of neural systems,* vol. 4, pp. 203-222, 1993

[xxxi] M. Mudrová and A. Procházka, "Principal

component analysis in image processing," in *Proceedings of the MATLAB Technical Computing Conference, Prague*, 2005.