Journal of Engineering and Applied Sciences 14 (2): 373-385, 2019
ISSN: 1816-949X

Deep Intelligent System for Human Recognition in Complex Domain

Swati Srivastava and Bipin Kumar Tripathi
Department of Computer Science and Engineering, Harcourt Butler Technical University, Kanpur, India

Abstract: This study aims to develop a deep computational model which is a novel aggregation of fuzzy clustering fused with evolutionary searching and a neural network based on a proposed artificial neuron structure in complex domain. In our Complex Deep Intelligent System (CDIS), we propose a complex neural classifier built upon a new complex neuron structure 'C-TROIKA'. The proposed deep model which is an amalgamation of Fused Fuzzy Distribution (FFD) and Complex Neural Classifier (CNC) capitulates an efficient tool for human recognition. The functional aptitudes of conventional neurons have been explored with complex-valued non-linear aggregation functions. This aggregation has the ability to confine higher-order correlations among input patterns. The proposed neuron structure based on these aggregation functions enables the system to provide faster convergence, better learning and recognition accuracy. The effectiveness and strengths of proposed complex neuron structure 'C-TROIKA' based deep intelligent system have been demonstrated over three benchmark biometric datasets, CASIA iris, Yale face and Indian face to realize the motivation.

Key words: Complex neuron structure, C-TROIKA, fused fuzzy distribution, complex neural classifier, effectiveness, intelligent system

INTRODUCTION

Over the years, researchers have developed a number of classification techniques in real domain with their own strengths and limitations (Wang *et al*., 2012; Oh *et al*., 2013; Lukas *et al*., 2016). Among them neural classifier is of great significance due to its efficient performance. Neural classifiers based on Real Back Propagation (R-BP) algorithm may stuck in local minima during learning which results in slow convergence and less accuracy. These limitations can be trounced by replacing the real numbers with the complex ones. Complex Back Propagation (C-BP) algorithm reduces the standoff probability in learning and significantly improves the convergence speed which has been recited by Nitta (1997) and Hirose (2006). Machine learning in complex domain has drawn considerable interest and attention from researches in the recent past Tripathi and Kalra (2011), Mandic and Goh (2009), Hirose (2006) where the outperformance of complex-valued neuron over real-valued neuron has been well stated. Tripathi and Kalra (2011) proposed a neuron model which incorporated an aggregation function based on the weighted root-power mean of complex-valued input signals. They used complex resilient propagation algorithm which provides better prediction accuracy and speedy training. To improve the performance of complex-valued back propagation neural networks, Chen *et al*. (2005) introduced a modified error function which added hidden layer error term in the conventional error. Error function plays a crucial role in performance evaluation of the complex-valued neural network as reported by Gangal *et al*. (2007) where it has been observed that for some error functions the performance depends on the architecture of the network whereas for some other error functions the convergence speed is independent of the network topology. Performance of complex neural network has been evaluated using different error functions by Gangal *et al*. (2007). In real domain, the ascendancy of deep architectures over traditional trivial architectures is evidenced by recent contributions (Zhang *et al*., 2017; Hong *et al*., 2017; Parkhi *et al*., 2015; Nagpal *et al*., 2015). Deep learning in complex domain still demands more attention to gain impressive advancements. In this study, the core idea is to acquire the benefits of complex domain embedded with that of deep architecture to develop a Complex Deep Intelligent System (CDIS) for human recognition. CDIS incorporates unsupervised fuzzy clustering fused with evolutionary searching and supervised neural classifier built on the complex novel neuron structure 'C-TROIKA'. It is worth mentioning here that the multiple computational intelligent techniques

Corresponding Author: Swati Srivastava, Department of Computer Science and Engineering, Harcourt Butler Technical University, Kanpur, India

embraced in our proposal are not competitive rather they are complementary to each other which has been demonstrated through an encouraging act of investigation of CDIS. In this study, we combine Principal Component Analysis (PCA) and Fisher Linear Discriminant (FLD) (Er *et al*., 2002) in real domain to extract features from input patterns. Then the mean input feature vectors undergo unsupervised fuzzy clustering eventually followed by evolutionary searching which enables the fuzzy algorithm to dig up the optimal partitioning. The upshot of unsupervised clustering is the cluster allocation matrix (γ) which imitates the cluster allocation to the input classes. This unsupervised segment is tracked by the supervised classification which is the job of complex-valued neural network possessing proposed complex neuron structure 'C-TROIKA'. An imaginary component of complex is added to the mean feature vectors which further act as a training set for the neural classifier. This complex neural classifier performs learning of complex-valued mean feature vectors based on γ obtained in previous segment.

MATERIALS AND METHODS

Complex deep intelligent system: In this study, we present a Complex Deep Intelligent System (CDIS) for human recognition which is robust to deal with large dataset of images possessing wide variations in features. The proposed system consists of two segments: Fused Fuzzy Distribution (FFD) which is unsupervised in nature and in real domain. This segment comprises of fuzzy c means clustering tracked by evolutionary search (Hruschka *et al*., 2009). It is accountable for optimal distribution of mean input feature vectors amongst number of clusters. Complex Neural Classifier (CNC) which is built on proposed complex C-TROIKA neurons and is liable for supervised classification in complex domain. This segment performs learning, based on the outcome of unsupervised segment and classification of input classes. Let us assume that C be the number of clusters and T denotes the maximum number of classes that can be allocated to each cluster. Fused fuzzy distribution segment results in C×T cluster allocation matrix (γ) where C and T are referred to as structural features because they decide the topology of the neural networks in the second segment of CDIS. The structural feature selection is one of the key challenges in deep learning (Angelov and Speroluti, 2016) which we have taken care of in our proposal. Assimilation of evolutionary algorithm in FFD segment assist the selection of structural parameters C (Tseng and Yang, 2001) and T. A neural

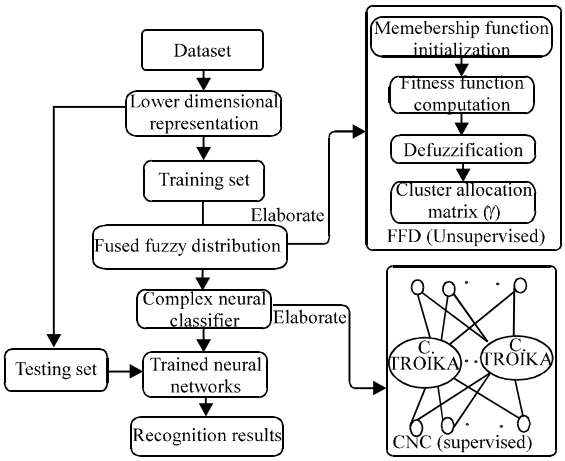

Fig. 1: General framework of proposed complex deep intelligent system

network is associated to each cluster and therefore, neural networks in second segment of CDIS are referred to as coupled higher order neural networks. As obvious due to above coupling, the number of coupled neural networks is equal to the number of clusters whilst the number of output neurons in each network is fixed at T. The complex neural classifier segment learns the input patterns in accordance with γ which act as the desired output for supervised segment. This learning is realized by the Complex Back Propagation (C-BP) algorithm with momentum which resolves the local minima problem (Nitta, 1997) unlike the Real Back Propagation (R-BP). As shown in Fig. 1, the proposed deep intelligent system includes image representation in lower dimension, fuzzy clustering algorithm fused with evolutionary search logic which is further complemented with Minkowsky distance measure followed by complex neural classifier.

Lower-dimensional feature space representation: The leading step in any image processing system is to find a suitable representation of images to be processed. The images with reduced amount of data though maintaining the adequate information for momentous learning is desirable representation. Unsupervised statistical approach such as Principal Component Analysis (PCA) or eigenface paradigm acquiesce a new representation with reduced data which contains only representative information. For better classification results, discriminant features of dataset are as well desirable which are extracted from Fisher Linear Disriminant (FLD) or discriminant eigenface paradigm (Er *et al*., 2002). FLD overcomes the limitations of PCA while retaining the idea of eigenface paradigm of projection from high dimensional feature space to a significantly lower dimensional feature

space. Thus, Eigenface pursued by fisherface method (Belhumeur *et al*., 1997) is used in the proposed Complex Deep Intelligent System (CDIS) to extract the desired features of input dataset.

Fused fuzzy distribution: This is an unsupervised segment of the proposed complex deep computational model which aims to obtain the cluster allocation matrix (γ). The processing of this segment begins with unsupervised clustering of input mean feature vectors in real domain through fuzzy c-means clustering. The optimal distribution of input classes cannot be obtained by only fuzzy c-means as for the same dataset it gives different partitions in different runs. To acquire the optimal partition which is required for further processing, evolutionary search strategy arrive into portrait. Here we refer different distributions, obtained by fuzzy c-means as populations. Evolutionary search is based on the concept of survival of the fittest which means the population with the largest fitness value will only survive. Based on the fitness function (Sheng *et al*., 2008) criteria, optimal distribution is searched by the evolutionary logic. Thus, the best partition among above gained dissimilar partitions is obtained by fuzzy clustering fused with evolutionary search which is then subject to defuzzification in order to attain γ. In this study as publicized in Fig. 1, we define steps of Fused Fuzzy Distribution (FFD) as the population initialization, fitness function based selection of the best population and defuzzification of the optimal partition.

Initialization of population: The process of allocating clusters to mean feature vectors of training classes starts with population initialization. The terms fuzzy partition matrix or membership function or population are used interchangeably in this study. Let $X = \{x_1, x_2, \ldots, x_p\}$ be the mean feature vectors of P classes. Fuzzy c-means clustering algorithm divides P classes into C clusters using fuzzy partition matrix U of size C×P. The initial population $U = (u_{ij})$ is randomly initialized such that:

$$u_{ij} \in [0,1]$$

where, $1 \le i \le C$ and $1 \le j \le P$. The population is updated by executing fuzzy clustering algorithm recursively until the objective function, described in next study, turn out to be minimized.

Fitness function based selection of the best population: In order to obtain a fuzzy partition matrix (a population), following objective function is iteratively minimized:

$$J = \sum_{i=1}^{C} \sum_{J=1}^{P} u_{ij}^{F} d_{ij} \quad (1)$$

where, $d_{ij} = \left(\sum_{k=1}^{p-1} \|x_{ik} - c_{ik}\|^{r}\right)^{r-1}$ is the distance of mean vector of jth class from the center of ith cluster. Here Minkowsky distance is used as a distance measure because of its ability to generate clusters having variable shapes depending on $\epsilon(1, \infty)$ known as generalization parameter. The underlying motivation behind using Minkowsky distance instead of Euclidean is its generalized nature as the shape of the cluster for the given problem primarily depends on the distance measure considered. The parameter F is weighting exponent which is known as fuzzifier, determines the degree of fuzziness and it lies in the range $(1, \infty)$. The selection of fuzzifier is imperative for fuzzy c-means implementation. The appropriate value of weighting exponent depends on data itself which has been shown by Yu *et al*. (2004). The process of minimizing the objective function requires following updates in membership function and cluster centers, respectively, Eq. 2:

$$u_{ij} = \left[\frac{d_{ij}^{\frac{-2}{F-1}}}{\sum_{k=1}^{C} d_{kj}}\right] \quad (2)$$

where, $1 \le i \le C$, $1 \le j \le P$ and Eq. 3:

$$c_i = \frac{\sum_{j=1}^{P} u_{ij}^{F} x_j}{\sum_{j=1}^{P} u_{ij}^{F}} \quad (3)$$

The termination of minimization process incorporates following condition:

$$\left\| J(u,c)^{t+1} - J(u,c)^{t} < \vartheta \right\| \quad (4)$$

where, $J(u, c)^t$ is the objective function at tth iteration and ϑ is pre-defined threshold. The fuzzy c-means clustering algorithm iteratively obtained number of dissimilar partitions. The evolutionary search is considered to find the best partition among the number of different partitions obtained. Let for r executions of fuzzy clustering algorithm we get r populations represented as $\{U_0, U_1, \ldots, U_r\}$. Fitness function based evolutionary search technique is used to find the best population among r populations where fitness function is expressed as:

$$f = \frac{Nd_{min}}{\sum_{i=1}^{C} \sum_{J=1}^{P} u_{ij}^{F} d} \quad (5)$$

where, $d = \sum_{i=1}^{c} \sum_{j=1}^{p} \|x_j - c_i\|^2$. The population with the highest fitness value is the optimal partition among different populations which is further used as initial

partition matrix for the next generation. The process is repeated for number of generations until the difference between partition matrix obtained for two successive generations is less than or equal to some defined threshold. The partition obtained in the last generation is the required optimal fuzzy partition U_{opt}.

To defuzzify the optimal distribution: The concluding step of unsupervised segment of the proposed deep model is to defuzzify U_{opt} which results in cluster allocation matrix (γ). In order to obtain the uniform structure of all clusters, we need to define the maximum cluster members (T). In U_{opt}, we sort all the classes in a cluster according to the degree of membership in descending order. Finally, γ is obtained by selecting top membership grade elements equal to T. Thus γ reveals the pre-classified clusters of the training data for which learning is performed in the neural network. Thus, γ act as the desired output in the proposed supervised classification segment. The parameter T participates in selecting output neurons in coupled neural networks. In empirical evaluation of the proposed complex model, we performed experiments with varying number of T keeping other parameters fixed and accounted the case which yielded plausibly good performance.

Complex neural classifier: The second segment of the proposed complex deep system is Complex Neural Classifier (CNC) which is responsible for supervised classification. An imaginary component of complex is added to the mean feature vectors of input classes which is considered to be the training set for this classifier. Thus, the training patterns are complex-valued where the imaginary part is negligible in comparison to the real counterpart. Let X be the mean feature vectors of input classes, then the complex input for the complex classifier will be X = X+I*0.001 where i is an imaginary unity. Now the coupled neural networks are trained for above complex inputs according to cluster allocation matrix (γ) obtained in previous clustering segment. We consider a frequently used three layer structure (L-M-N){C} for coupled networks. First layer has L = P-1 inputs, second layer has M proposed complex neurons C-TROIKA, third layer consists of N = T complex conventional neurons and C is the number of coupled neural networks. Complex proposed neurons C-TROIKA are computational efficient which ensures the improved convergence speed and prediction precision of proposed complex neural classifier than C-MLP. Here, all inputs and weights are considered to be complex numbers. Conventionally, w_{lm} represents the weight from lth neuron to mth neuron. Let input vector be $Z = \{z_1, z_2, \ldots, z\}$, $WS_m = \{ws_{1m}, ws_{2m}, \ldots, ws_{LM}\}$ be the weights from inputs to the summation part of m^{th} proposed complex neuron and $WRB_m = \{wrb_{1m}, wrb_{2m}, \ldots, wrb_{Lm}\}$ be the weights from inputs to the radial basis part of m^{th} neuron. Let $W_0 = \{w_{01}, w_{02}, \ldots, w_{0M}\}$ be the bias weight vector and $zm_0 = 1+I*0.001$ be the bias input for M complex C-TROIKA neurons in the hidden layer. Let $Wn = \{w_{1n}, w_{2n}, \ldots, w_{Mn}\}$ be the weight vector of hidden neurons to nth output neuron, $X_0 = \{x_{01}, x_{02}, \ldots, x_{0N}\}$ be the bias weight vector and $zn_0 = 1+i*0.01$ be the bias input for N complex conventional neurons in the output layer. The (.) represents the complex conjugation and $(.)^T$ represents the matrix transposition. The imprecision involved will be taken care by two compensatory parameters Θ and Ω which stipulates the summation and radial basis contributions, respectively, associated with each hidden neuron. In this study, R and ξ represents the real and imaginary components of complex, respectively. The net potential of mth C-TROIKA neuron in hidden layer can be figured through generalized product Eq. 6:

$$\begin{aligned} V_m &= R(V_m)+i\xi(V_m) \\ V_m &= V_{m1}+V_{m2}+V_{m1}V_{m2}+w_{0m}zm_0 \end{aligned} \quad (6)$$

where:

$$V_{m1} = T_m WS_m Z^T$$

$$V_{m2} = O_m \exp\left(-\left\|Z - WRB_m\right\|^2\right)$$

The output of mth C-TROIKA neuron can be expressed as Eq. 7:

$$Y_m = f_c(V_m) = R(Y_m)+i\xi(Y_m) \quad (7)$$

where f_c is complex-valued activation function whose selection depends on the type of application. The net potential and output of nth complex conventional neuron in output layer, respectively can be given by:

$$\begin{aligned} V_n &= R(V_n)+i\xi(V_n) \\ V_n &= \sum_{m=1}^{M} w_{mn}Y_m + x_{0n}zm_0 \end{aligned} \quad (8)$$

and Eq. 9:

$$Y_n = f_c(V_n) = R(Y_n)+i\xi(Y_n) \quad (9)$$

Let Y_d be the desired output, then the error at nth output neuron can be computed as:

$$e_n = R(e_n)+i\xi(e_n) = Y_{dn} - Y_n \quad (10)$$

The complex-valued cost function (MSE) can be given by:

$$E = \frac{1}{2}\sum_{n=1}^{N} e_n \overline{e_n} = \frac{1}{2}\sum_{n=1}^{N}\left[\left(R(e_n)\right)^2 + \left(\xi(e_n)\right)^2\right] \quad (11)$$

Weight update rules: For efficient training, minimization of cost function is required which can be realized by the Complex Back Propagation (C-BP) algorithm with momentum which amend the weights coefficient recursively till the cost becomes minimized. This weight amendment is given by:

$$w_{new} = w_{old} + \Delta w^{(t)} + \alpha \Delta w^{(t-1)} \quad (12)$$

where Eq. 13:

$$\begin{aligned} \Delta w &= -\eta \nabla_w E \\ \Delta w &= -\eta \nabla_{R(w)} E - i\eta \nabla_{\xi(w)} E \\ \Delta w &= -\eta\left(\frac{\partial E}{\partial R(w)} + i * \frac{\partial E}{\partial \xi(w)}\right) \end{aligned} \quad (13)$$

Where:

η	=	The learning rate
α	=	The momentum factor whose values ranges in between 0 and 1
t	=	The iteration
$\nabla_{R(w)}$ and $\nabla_{\xi(w)}E$	=	The gradients with respect to real and imaginary parts of complex weights, respectively
Δ	=	Initialized to zero

Let f_c be the derivative of the complex-valued activation function f_c. The weight update equation between hidden and output layer, for any weight $w = w_{mn}$ in output layer, can be obtained by Eq. 13 as Eq. 14:

$$\Delta w_{mn} = -\eta\left(\frac{\partial E}{\partial R(w_{mn})} + i * \frac{\partial E}{\partial \xi(w_{mn})}\right) \quad (14)$$

Where Eq. 15:

$$-\frac{\partial E}{\partial R(w_{mn})} = \begin{bmatrix} R(e_n) f_c'\left(R(V_n)\right)\dfrac{\partial R(V_n)}{\partial R(w_{mn})} \\ +\xi(e_n) f_c'\left(\xi(V_n)\right)\dfrac{\partial \xi(V_n)}{\partial R(w_{mn})} \end{bmatrix} \quad (15)$$

And Eq. 16:

$$-\frac{\partial E}{\partial \xi(w_{mn})} = \begin{bmatrix} R(e_n) f_c'\left(R(V_n)\right)\dfrac{\partial R(V_n)}{\partial \xi(w_{mn})} \\ +\xi(e_n) f_c'\left(\xi(V_n)\right)\dfrac{\partial \xi(V_n)}{\partial \xi(w_{mn})} \end{bmatrix} \quad (16)$$

Substituting Eq. 15 and 16 in Eq. 14, we get Eq. 17:

$$\begin{aligned} \Delta w_{mn} &= \eta\begin{bmatrix} R(e_n) f_c'\left(R(V_n)\right)\left\{\dfrac{\partial R(V_n)}{\partial R(w_{mn})} + i\dfrac{\partial R(V_n)}{\partial \xi(w_{mn})}\right\} \\ +\xi(e_n) f_c'\left(\xi(V_n)\right)\left\{\dfrac{\partial \xi(V_n)}{\partial R(w_{mn})} + i\dfrac{\partial \xi(V_n)}{\partial \xi(w_{mn})}\right\} \end{bmatrix} \\ \Delta w_{mn} &= \eta\overline{Y_m}\left(R(e_n) f_c'\left(R(V_n)\right) + i\xi(e_n) f_c'\left(\xi(V_n)\right)\right) \end{aligned} \quad (17)$$

Bias update Eq. 18:

$$\Delta x_{0n} = \eta\overline{z_{n_0}}\left(R(e_n) f_c'\left(R(V_n)\right) + i\xi(e_n) f_c'\left(\xi(V_n)\right)\right) \quad (18)$$

The weight update equation between input and hidden layer, for any weight $w = w_{lm}$ can be generalized as Eq. 19:

$$\Delta w_{lm} = -\eta\left(\frac{\partial E}{\partial R(w_{lm})} + i * \frac{\partial E}{\partial \xi(w_{lm})}\right) \quad (19)$$

Where Eq. 20:

$$\begin{aligned} -\frac{\partial E}{\partial R(w_{lm})} &= \frac{\partial R(V_m)}{\partial R(w_{lm})} f_c'\left(R(V_m)\right)\sum_{n=1}^{N} \\ &\left\{\begin{matrix} R(e_n) f_c'\left(R(V_n)\right) R(w_{mn}) + \\ \xi(e_n) f_c'\left(\xi(V_n)\right)\xi(w_{mn}) \end{matrix}\right\} \\ &+\frac{\partial(\xi(V_m))}{\partial R(w_{lm})} f_c'\left(\xi(V_m)\right)\sum_{n=1}^{N}\left\{\begin{matrix} \xi(e_n) f_c'\left(\xi(V_n)\right) R(w_{mn}) - \\ R(e_n) f_c'\left(R(V_n)\right)\xi(w_{mn}) \end{matrix}\right\} \end{aligned} \quad (20)$$

Similarly, $\partial E/\partial \xi(w_{lm})$ can be expressed. To obtain the weight update equation for summation aggregation of C-TROIKA neurons at hidden layer, Eq. 20 can be rewritten as:

$$\begin{aligned} -\frac{\partial E}{\partial R(ws_{lm})} &= \frac{\partial R(V_m)}{\partial R(ws_{lm})} R(K_m) + \frac{\partial \xi(V_m)}{\partial R(ws_{lm})}\xi(K_m) \\ -\frac{\partial E}{\partial R(ws_{lm})} &= R(K_m) \\ &\left\{\begin{matrix} \left(R(\Theta_m) + R(\Theta_m) R(V_{m2}) - \xi(\Theta_m)\xi(V_{m2})\right) R(z_1) \\ -\left(\xi(\Theta_m) + \xi(\Theta_m) R(V_{m2}) + R(\Theta_m)\xi(V_{m2})\right)\xi(z_1) \end{matrix}\right\} \\ &+\xi(K_m)\left\{\begin{matrix} \left(\xi(\Theta_m) + \xi(\Theta_m) R(V_{m2}) + R(\Theta_m)\xi(V_{m2})\right) R(z_1) \\ +\left(R(\Theta_m) + R(\Theta_m) R(V_{m2}) - \xi(\Theta_m)\xi(V_{m2})\right)\xi(z_1) \end{matrix}\right\} \end{aligned} \quad (21)$$

And Eq. 22:

$$-\frac{\partial E}{\partial \xi(ws_{lm})} = \frac{\partial R(V_m)}{\partial ?(ws_{lm})} R(K_m) + \frac{\partial \xi(V_m)}{\partial \xi(ws_{lm})} \xi(K_m)$$

$$-\frac{\partial E}{\partial \xi(ws_{lm})} = R(K_m)$$

$$\left\{ \begin{pmatrix} -R(\Theta_m) + R(\Theta_m) R(V_{m2}) \\ -\xi(\Theta_m)\xi(V_{m2}) \end{pmatrix} \xi(z_l) - \right.$$

$$\left. \begin{pmatrix} \xi(\Theta_m) + \xi(\Theta_m) \\ R(V_{m2}) + R(\Theta_m)\xi(V_{m2}) \end{pmatrix} R(z_l) \right\}$$

$$+\xi(K_m) \begin{Bmatrix} \left(R(\Theta_m) + R(\Theta_m) R(V_{m2}) - \xi(\Theta_m)\xi(V_{m2})\right)\Theta(z_l) \\ -\left(\xi(\Theta_m) + \xi(\Theta_m) R(V_{m2}) + R(\Theta_m)\xi(V_{m2})\right)\xi(z_l) \end{Bmatrix} \quad (22)$$

Thus, by substituting Eq. 21 and 22 in Eq. 19, we get Eq. 23:

$$\Delta ws_{lm} = \eta \begin{Bmatrix} \left(R(z_l) - i\xi(z_l)\right)\left(R(\Theta_m) - i\xi(\Theta_m)\right) \\ \left(1 + R(V_{m2}) - i\xi(V_{m2})\right)\left(R(K_m) + i\xi(K_m)\right) \end{Bmatrix}$$

$$\Delta ws_{lm} = \eta \overline{z_i \Theta_m} \left(1 + \overline{V_{m2}}\right) k_m \quad (23)$$

where:

$$R(K_m) = f_c'\left(R(V_m)\right)$$

$$\sum_{n=1}^{N} \begin{Bmatrix} R(e_n) f_c'\left(R(V_n)\right) R(w_{mn}) \\ +\xi(e_n) f_c'\left(\xi(V_n)\right)\xi(w_{mn}) \end{Bmatrix}$$

and:

$$(K_m) = f_c'\left(\xi(V_m)\right)$$

$$\sum_{n=1}^{N} \begin{Bmatrix} \xi(e_n) f_c'\left(\xi(V_n)\right) \\ R(w_{mn}) - R(e_n) f_c'\left(R(V_n)\right)\xi(w_{mn}) \end{Bmatrix}$$

Similarly, the weight updates for radial basis aggregation, compensatory weights associated with summation and radial basis contributions and bias of proposed C-TROIKA neurons can be, respectively articulated as:

$$wrb_{lm} = 2\eta \exp\left(-\|Z - WRB_m\|^2\right)(z_l - wrb_{lm})$$

$$\times \begin{bmatrix} R(K_m)\left\{R(\Omega_m)\left(1 + R(V_{m1})\right) - \xi(\Omega_m)\xi(V_{m1})\right\} \\ +\xi(K_m)\left\{\xi(\Omega_m)\left(1 + R(V_{m1})\right) + \xi(\Omega_m)\xi(V_{m1})\right\} \end{bmatrix} \quad (24)$$

$$\Delta\Theta_m = \eta\left(WS_m Z^T\right)\left(1 + \overline{V_{m2}}\right)K_m \quad (25)$$

$$\Delta\Omega_m = \eta \exp\left(-\|Z - WRB_m\|^2\right)\left(1 + \overline{V_{m1}}\right)K_m \quad (26)$$

$$\Delta w_{0m} = \eta \overline{zm_0} K_m \quad (27)$$

The updated weights can be obtained from Eq. 12. This process of updating weights recursively continues till the cost becomes minimized and proposed complex C-TROIKA based classifier is trained in accordance with the desired output γ.

Recognition: Once we get trained neural networks, testing can be performed by using the function Max of Max of the outputs of each coupled neural network. For every input test pattern, we have obtained the cxl output matrix. Let O_k (NN_i) be the maximum output of ith coupled NN for kth testing pattern. The resulting class of a test pattern can be identified by finding the cluster corresponding to $Max^c{}_{i=1}$ (O_k (NN_i)).

RESULTS AND DISCUSSION

In order to evaluate the performance of the proposed complex deep intelligent system, experiments are conducted on the 3 benchmark biometric datasets-CASIA Iris, Yale Face and Indian Face. The recognition results for all 3 databases are accounted for number of parameters which includes number of Clusters (C), maximum cluster members (T) and number of proposed C-TROIKA hidden neurons. Performance is recorded by varying one parameter keeping rest of the parameters constant for C-MLP as well as for proposed C-TROIKA based complex classifier. The performance of both the above mentioned classifiers is compared in terms of the number of hidden neurons and their corresponding accuracy. Performance of the proposed complex deep model is also compared with recent existing methodologies. Furthermore, standard biometric measures such as FAR (False Acceptance Rate) and FRR (False Rejection Rate) are recorded to the evaluate the performance of the proposed model.

Evaluation on CASIA Iris dataset: CASIA Iris dataset was collected by Chinese Academy of Science's Institute of Automation. The development of the dataset had been started from CASIA Iris-V1 which was the first version and updated up to CASIA Iris-V3. CASIA Iris-V4 is an extension of CASIA Iris-V3 and contains six subsets. The three subsets are from CASIA Iris-V3 which includes CASIA Iris-Interval, CASIA Iris-Lamp and CASIA Iris-twins and remaining three are new subsets which includes CASIA Iris-Distance, CASIA Iris-1000 and CASIA Iris-Syn. CASIA Iris-V4 contains 54601 iris images from more than 1800 subjects. In our experiments we have used CASIA Iris-Interval subset of CASIA Iris-V4 which

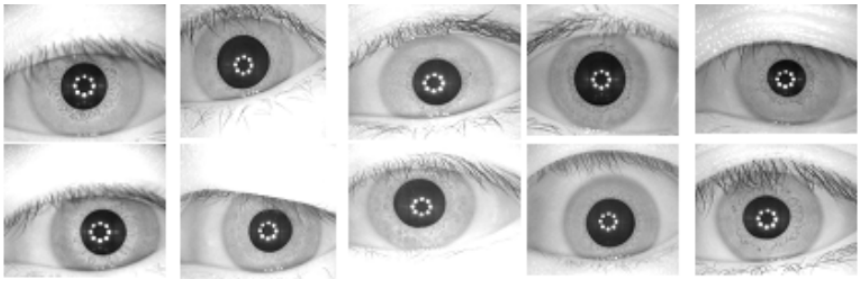

Fig. 2: CASIA iris interval-V4 sample images

Table 1: Recognition accuracy for CASIA Iris dataset

C-TROIKA neurons	C	T	MSE	Accuracy (%) Training set	Testing set
5	6	8	0.02221	98.0	88.6
		10	0.02311	98.0	91.3
		12	0.03998	98.0	91.0
	8	8	0.02556	98.5	90.5
		10	0.02654	99.0	94.6
		12	0.03874	99.0	94.3
	10	8	0.03825	99.5	89.8
		10	0.04082	99.5	94.2
		12	0.05963	99.6	93.6
7	6	8	0.00202	100	90.6
		10	0.00289	100	94.5
		12	0.00336	100	94.0
	8	8	0.00119	100	94.7
		10	**0.00231**	**100**	**99.8**
		12	0.00409	100	97.6
	10	8	0.00325	100	94.2
		10	0.00422	100	97.5
		12	0.00559	99.8	97.1
9	6	8	0.00391	99.9	90.2
		10	0.00479	99.9	94.2
		12	0.00516	99.0	93.6
	8	8	0.00284	99.8	94.5
		10	0.00368	99.8	99.2
		12	0.00423	99.8	97.0
	10	8	0.03247	99.6	93.9
		10	0.05248	99.8	97.2
		12	0.06663	99.8	96.9

Bold values are significant

Table 2: Comparison based on neuron type for CASIA Iris dataset

Neuron type/ Hidden neurons	C	T	MSE	Accuracy(%) Training set	Testing set
C-MLP					
12	6	10	0.05909	96.8	90.9
		12	0.06112	96.6	90.7
	8	10	0.03994	98.2	94.8
		12	0.04214	98.2	94.2
	10	10	0.06973	98.8	94.5
		12	0.07916	98.8	94.0
C-TROIKA					
5	6	10	0.02311	98.0	91.3
		12	0.03998	98.0	91.0
	8	10	0.02654	99.0	94.6
		12	0.03874	99.0	94.3
	10	10	0.04082	99.5	94.2
		12	0.05963	99.6	94.1
C-MLP					
15	6	10	0.04562	98.9	94.5
		12	0.05968	98.8	94.0
	8	10	0.05113	99.9	99.2
		12	0.06333	99.9	97.0
	10	10	0.06242	99.9	97.1
		12	0.07901	99.8	96.6
C-TROIKA					
7	6	10	0.00289	100	94.5
		12	0.00336	100	94.2
	8	**10**	**0.00231**	**100**	**99.8**
		12	0.00409	100	97.5
	10	10	0.00422	100	97.5
		12	0.00559	99.8	97.2

Bold values are significant

contains images of 249 subjects. In this study we have considered 50 subjects, each consists of 20 images including left and right iris images. For training, 7 images of each subject are used and remaining 13 images are used to test the performance of proposed deep system. Fig. 2. shows some sample images from this dataset.

The performance of the proposed deep model for this database is revealed in Table 1 where it can be clearly observed that the testing accuracy varies with the variation in different parameters. C-TROIKA = 7, number of Clusters C = 8 and maximum cluster members T = 10 constitutes a set of parameter values at which highest accuracy is achieved which is 99.8%.

In order to compare C-TROIKA based classifier with C-MLP, we have recorded the accuracy of our model for this database using both the classifiers. In our experiments, for same set of values of C and T, testing accuracy is accounted for different number of hidden neurons both for proposed classifier and C-MLP. It can be evidenced from Table 2 that C-TROIKA based classifier outperforms C-MLP as it gained the approximate accuracy as that of C-MLP with lesser number of hidden neurons which reduces the computational complexity. The corresponding graph is depicted in Fig. 3 where it can be evidently observed that C-MLP requires adequately larger number of hidden neurons as compared to proposed C-TROIKA based classifier. Also, the better performance of our classifier than C-MLP in the context of speedy convergence is visibly examined in Fig. 4. The comparison of proposed Complex Deep Intelligent System (CDIS) with recent methodologies is summarized in Table 3 which attested the better performance of the proposed model. For the same set of other parameters, the FAR and FRR

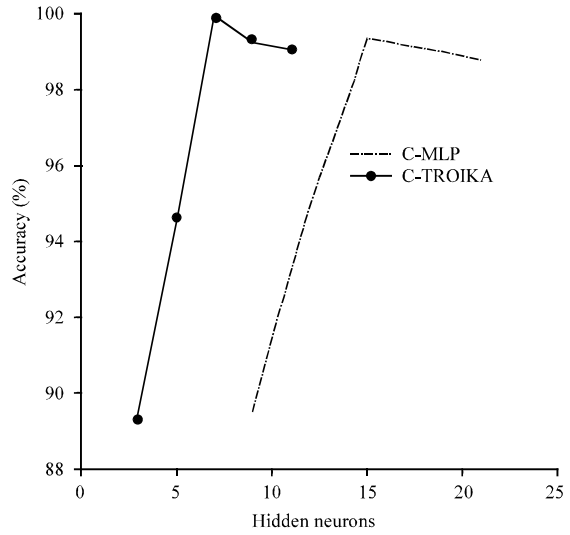

Fig. 3: Comparison of proposed classifier (C-TROIKA neurons in hidden layer) with C-MLP (complex conventional neurons in hidden layer) for CASIA iris dataset

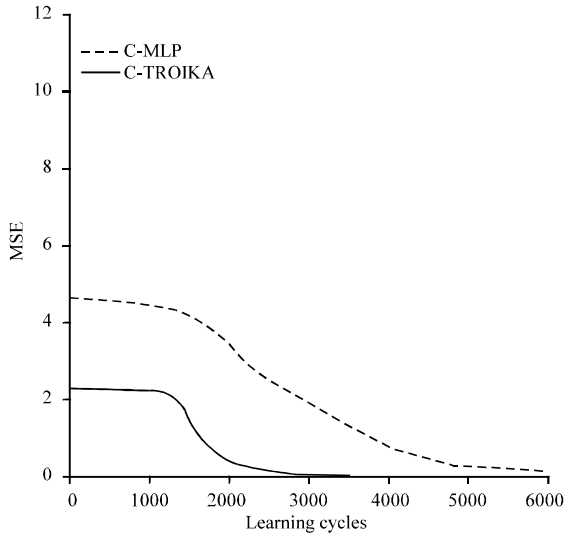

Fig. 4: Error convergence graph for CASIA iris interval

Table 3: Comparison with other methodologies for CASIA Iris dataset

Methods	Recognitionrate (%)
GEFE-MLBP (O'Connor *et al*., 2014)	91.17
Texture feature extraction method (Hajari *et al*., 2016)	94.5
Daugman (2001)	98.60
CDIS (Proposed)	99.8

for proposed C-TROIKA based classifier is recorded as 0.03 and 3.2%, respectively and that of C-MLP is 0.56 and 4.7%. These recorded values of FAR and FRR demonstrates the aptitude of proposed C-TROIKA complex neuron structure.

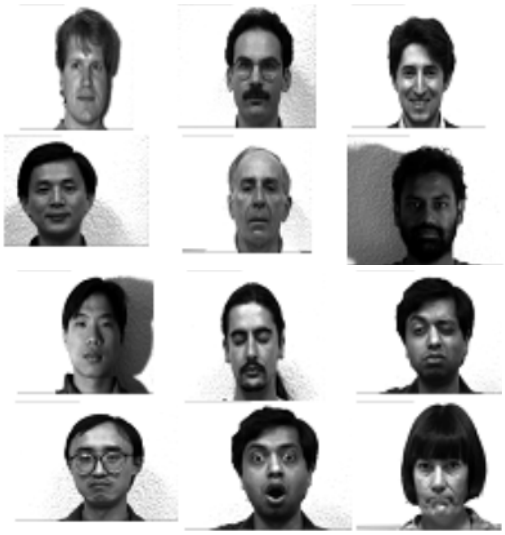

Fig. 5: sample images from Yale face dataset

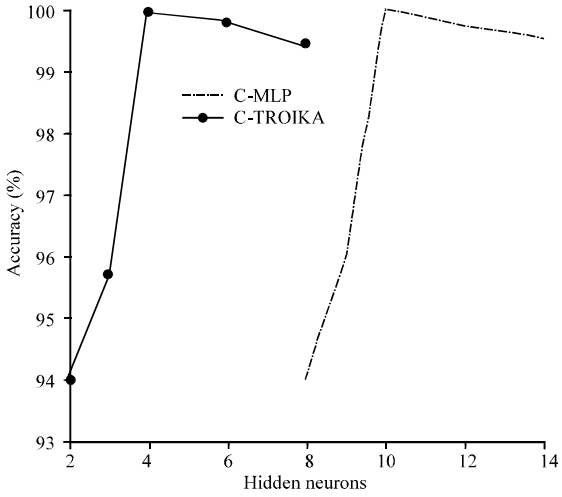

Fig. 6: Comparison of proposed classifier (C-TROIKA neurons in hidden layer) with C-MLP (complex conventional neurons in hidden layer) for Yale face dataset

Evaluation on Yale face dataset: This database contains 165 face images of 15 subjects with 11 images per subject. Each of the images per person is having different expression or configuration such as center-light with glasses, happy, left-light without glasses, normal, right-light, sad, sleepy, surprised and wink. All images are grayscale and in GIF format. In this study, we use 5 images per subject for training and 6 images per person to train the model. Few sample images are shown in Fig. 5 and 6.

The accuracy results of Yale faces are exemplified in Table 4 for different sets of parameters. The best recognition accuracy is 100% which is achieved at

Table 4: Recognition accuracy for yale face dataset

C-TROIKA Neurons	C	T	MSE	Accuracy (%) Training set	Testing set
2	5	6	0.02325	96.2	85.3
		8	0.03557	97.2	90.8
		10	0.04468	97.4	90.5
	7	6	0.02598	98.0	89.2
		8	0.03676	98.3	93.9
		10	0.05936	99.0	93.4
	9	6	0.02111	99.0	89.0
		8	0.03898	99.0	93.8
		10	0.04776	99.0	93.5
4	5	6	0.00421	99.0	92.3
		8	0.00556	99.0	97.2
		10	0.00662	99.5	97.0
	7	6	0.00633	99.9	94.8
		8	**0.00789**	**100**	**100**
		10	0.00811	100	99.7
	9	6	0.00932	100	94.3
		8	0.07633	100	99.5
		10	0.06212	100	98.3
6	5	6	0.04979	100	92.0
		8	0.05753	100	96.9
		10	0.06448	100	96.8
	7	6	0.03254	100	97.5
		8	0.04885	100	99.8
		10	0.05326	100	99.6
	9	6	0.02357	100	94.1
		8	0.03961	100	99.2
		10	0.04901	100	98.1

Table 5: Comparison based on neuron type for yale face dataset

Neuron type Hidden	C	T	MSE	Accuracy (%) Training set	Testing set
C-MLP					
8	5	8	0.03480	97.8	91.0
		10	0.04406	98.0	90.7
	7	8	0.05294	99.9	94.0
		10	0.06479	99.9	93.8
	9	8	0.07226	99.9	93.9
		10	0.08994	99.9	93.2
C-TROIKA					
2	5	8	0.03557	97.2	90.8
		10	0.04468	97.4	90.5
	7	8	0.03676	98.3	93.9
		10	0.05936	99.0	93.4
	9	8	0.03898	99.0	93.8
		10	0.04776	99.0	93.5
C-MLP					
10	5	8	0.05809	99.2	97.4
		10	0.06255	99.9	97.2
	7	8	0.07687	100	100
		10	0.09967	100	99.8
	9	8	0.07431	100	98.8
		10	0.08865	100	98.4
C-TROIKA					
4	5	8	0.00556	99.0	97.2
		10	0.00662	99.5	97.0
	7	**8**	**0.00789**	**100**	**100**
		10	0.00811	100	99.7
	9	8	0.07633	100	99.5
		10	0.06212	100	99.5

Bold values are significant

C-TROIKA = 4, C = 7 and T = 8. Classifier based on C-TROIKA neurons gives efficiently better recognition accuracy than C-MLP with smaller network topology for the same set of other parameters which is illustrated in Table 5. The number of hidden neurons is a measure of the great essence for comparisons of the two aforesaid classifiers. The plot in Fig. 6 represents the comparisons based on the number of proposed neurons C-TROIKA and complex conventional neurons in the hidden layer of neural network from where it can be stated that the proposed C-TROIKA based classifier requires lesser number of hidden neurons than C-MLP to gain the estimated same accuracy. The C-TROIKA based classifier provides faster convergence than C-MLP which can be evidenced by the error convergence graph as shown in Fig. 7 for the same set of parameters. The proposed C-TROIKA neurons facilitates the recognition system to provide reduced computational complexity (due to compact network structure) and rapid convergence than C-MLP which can be attested from Fig. 6 and 7, respectively. Proposed model achieves the best recognition accuracy of 100% for this database which is highest among recent state-of-the-art methods as shown in Table 6. The standard biometric measures FAR and FRR for this database is traced as 0.01 and 3.8% for the proposed classifier, respectively and for the same set of parameters the corresponding FAR and FRR for C-MLP is 0.09 and 4.6%, respectively. The above values of FAR and FRR again uphold the supremacy of C-TROIKA neurons over complex conventional neurons.

Table 6: Comparison with other methodologies for yale face dataset

Method	Recognition Accuracy (%)
OCON Classifier (Tripathi, 2017)	97.4
McDFR (Chen *et al*., 2015)	97.78
McMmFL (Aslan *et al*., 2017)	98.97
CDIS (Proposed)	100

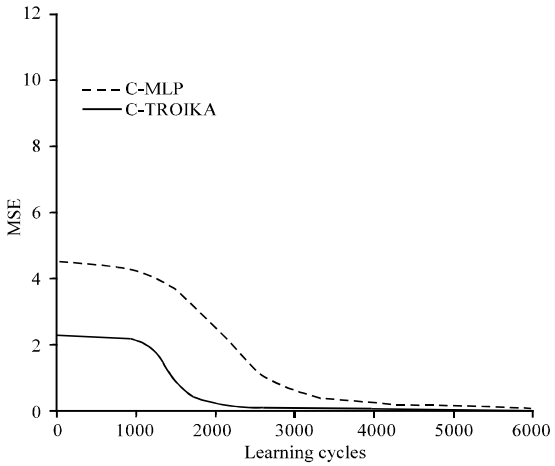

Fig. 7: Error convergence graph for Yale dataset

Evaluation on Indian face dataset: This database was introduced by Jain and Mukherjee in 2002 which contains the face images of female and male subjects with different orientations. Each subject consists of 10 images with wide

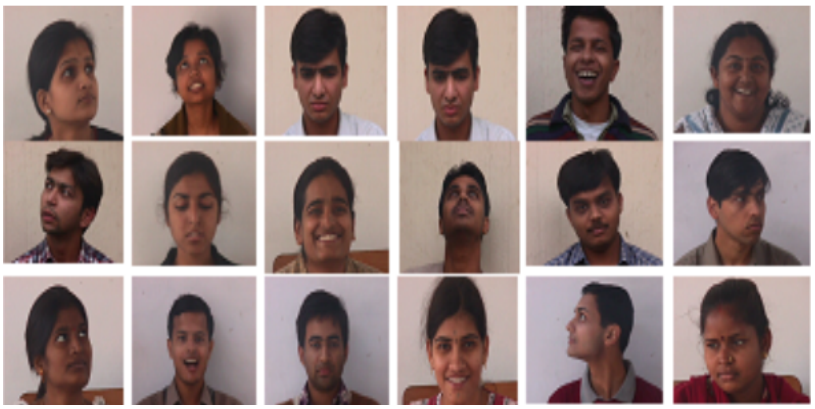

Fig. 8: Sample images from Indian face dataset

Table 7: Recognition accuracy for indian face dataset

C-TROIKA				Accuracy (%)	
Neurons	C	T	MSE	Training set	Testing set
4	8	12	0.00707	96.2	85.2
		14	0.00843	97.2	89.4
		16	0.00998	97.0	89.1
	10	12	0.00548	97.2	90.4
		14	0.00699	98.2	94.3
		16	0.00724	98.0	94.1
	12	12	0.00583	96.9	90.2
		14	0.00856	98.0	94.0
		16	0.00932	98.0	93.8
6	8	12	0.00633	98.6	88.3
		14	0.00802	99.0	90.8
		16	0.00964	98.9	90.2
	10	12	0.00494	99.8	95.6
		14	**0.00532**	**99.9**	**96.9**
		16	0.00628	99.9	95.2
	12	12	0.00594	99.8	95.0
		14	0.00752	99.9	96.1
		16	0.00824	99.8	95.0
8	8	12	0.00683	98.6	88.0
		14	0.00822	98.8	90.2
		16	0.00973	99.0	90.0
	10	12	0.00477	99.8	94.2
		14	0.00587	99.8	96.5
		16	0.00611	99.8	94.9
	12	12	0.00688	99.6	94.8
		14	0.00813	99.8	95.6
		16	0.00966	99.8	94.5

Bold values are significant

Table 8: Comparison based on Neuron type for Indian face dataset

				Accuracy (%)	
Neuron type/Hidden	C	T	MSE	Training set	Testing set
C-MLP					
10	8	14	0.04698	96.8	89.5
		16	0.06323	96.6	89.0
	10	14	0.05981	98.5	94.5
		16	0.07260	98.5	94.0
	12	14	0.08023	98.8	94.2
		16	0.09753	98.8	93.5
C-TROIKA					
4	8	14	0.00843	97.2	89.4
		16	0.00998	97.0	89.1
	10	14	0.00699	98.2	94.3
		16	0.00724	98.0	94.1
	12	14	0.00856	98.0	94.0
		16	0.00932	98.0	93.8
C-MLP					
15	8	14	0.08589	98.9	90.5
		16	0.01225	98.8	90.0
	10	14	0.06024	99.9	96.8
		16	0.07369	99.9	96.2
	12	14	0.07431	99.9	95.9
		16	0.09865	99.8	95.7
C-TROIKA					
6	8	14	0.00802	99.0	90.8
		16	0.00964	98.9	90.2
	10	14	0.00532	99.9	96.9
		16	0.00628	99.9	95.2
	12	14	0.00752	99.9	96.1
		16	0.00824	99.8	95.0

variations. All images have a bright homogenous background, different poses including left, right, up, down, up-right, up-left and different emotions. Figure 8 shows the variations in images of dataset. Here, we considered 50 subjects where 4 images per subject are randomly selected for training and rest of the images are used as a testing set, thus, the system has trained by 200 images and rest 300 images are used to test the system.

The recognition results for Indian database are presented in Table 7 where accuracies are recorded for different sets of parameters. The variation in parameters affects the accuracy of the model. The best accuracy has attained at C-TROIKA = 6, C = 10 and T = 14. Table 8 depicts the comparison of C-TROIKA based classifier with C-MLP based on the number of hidden neurons. The proposed complex classifier is computationally efficient as it requires moderately lesser number of C-TROIKA neurons than complex conventional neurons in C-MLP to arrive at the approximate same accuracy. The corresponding graph is shown in Fig. 9 where the nearly similar accuracy graph is obtained for the two aforesaid classifiers at different number of hidden neurons.

From Fig. 10. it is observed that the proposed classifier based on C-TROIKA offers better rate of convergence than C-MLP which again demonstrates the

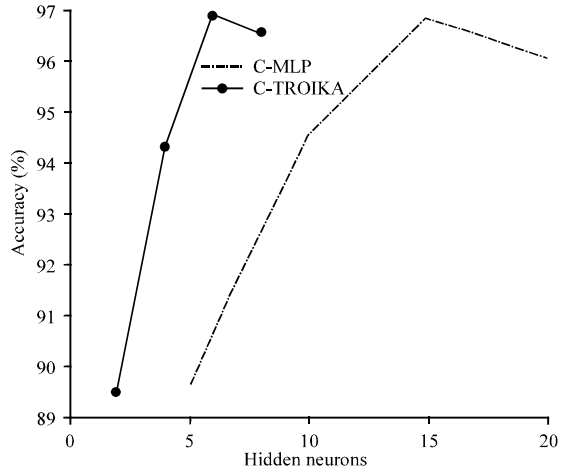

Fig. 9: Comparison of proposed classifier (C-TROIKA neurons in hidden layer) with C-MLP (complex conventional neurons in hidden layer) for Indian face dataset

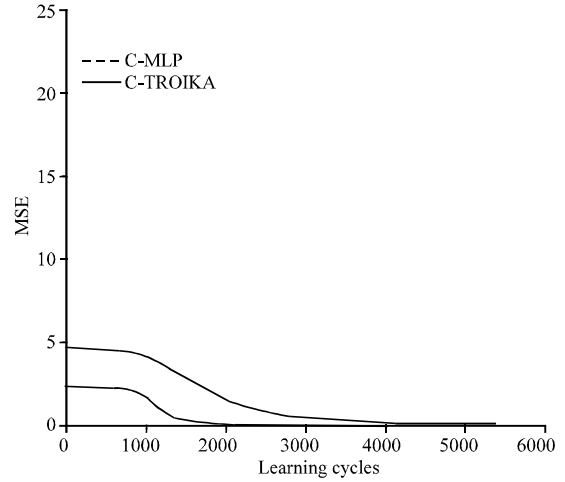

Fig. 10: Error convergence graph for Indian face dataset

Table 9: Comparison with other methodologies for indian face dataset

Method	Recognition accuracy (%)
ANFIS Classifier (Arivazhagan *et al*., 2014)	90.47
OCON Classifier (Tripathi, 2017)	95.4
CDIS (Proposed)	96.9

alluring performance of C-TROIKA neurons. The comparison of proposed complex deep model with existing techniques is abridged in Table 9 which confirms that our proposal offers superior performance. FAR for proposed complex classifier is acquired as 0.08% and the corresponding FRR is 4.1% whereas for C-MLP FAR is 0.24% and FRR is obtained as 5.2%. The lower values of FAR and FRR of our classifier again designates the improved performance of proposed C-TROIKA neuron over complex conventional neuron.

This study presents a deep computational model which is a novel synergism of fused fuzzy distribution segment in real domain and proposed C-TROIKA neuron model based neural classifier segment in complex domain. Extensive experiments conducted in this study demonstrate that the proposed model endows with superior recognition accuracy and simultaneously robust to unauthorized cases. The FRRs are recorded at very low FARs for all three databases which makes our proposed recognizer strict for both the authorized and unauthorized persons. Performance evaluation presented in this study reported the enthralling performance of our deep intelligent system which attested the beauty of C-TROIKA neuron. The remarkable triumph in the proposed complex classifier is its compact topology, enhanced convergence rate and better prediction accuracy.

We have performed all our experiments also in real domain where we have reached the conclusion that for the same set of parameters, complex domain results are far better in terms of accuracy, convergence, training and number of learning cycles than real domain results. Further, in complex domain, comparisons have been made between C-MLP and proposed C-TROIKA neuron based neural network. The experimental results and comparisons in literature review verified that the classifier based on proposed C-TROIKA provides enhanced performance than C-MLP where the number of hidden neurons used in the network topology is the key parameter of assessment.

The experiments performed on three benchmark biometric datasets-CASIA Iris, Yale face, Indian face for the proposed CDIS Model demonstrated its out performance over the complex conventional neuron based hybrid system. From the results presented in Tables 1, 4 and 7 one can observe the affect of variations in the number of hidden neurons (C-TROIKA), Clusters (C) and maximum cluster members (T) on the testing accuracy of the system. Figure 11 and 12 exhibits these variation affects more clearly. In our experiments for CASIA Iris dataset, best accuracy is recorded when C-TROIKA is 7 whereas for Yale and Indian face datasets best results are obtained at C-TROIKA equal to 4 and 6, respectively. Significant improvement in accuracy is observed up to certain number of hidden neurons for constant T and C, after that no prominent improvement is recorded. Figures 3, 6 and 9 reveals the plots which confirms the above statement. For fixed number of hidden neurons and maximum cluster members, one can monitor the affect of number of clusters from Table 1,4 and 7 and Fig. 11 for the three databases. For CASIA Iris, Yale and Indian datasets, we observe that recognition rate enhances up to C = 8, C = 7 and C = 10, respectively after which there is

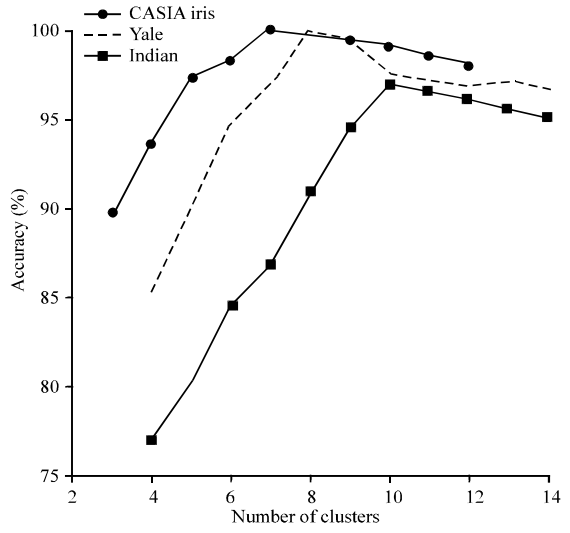

Fig. 11: Impact on accuracy by varying number of clusters (C)

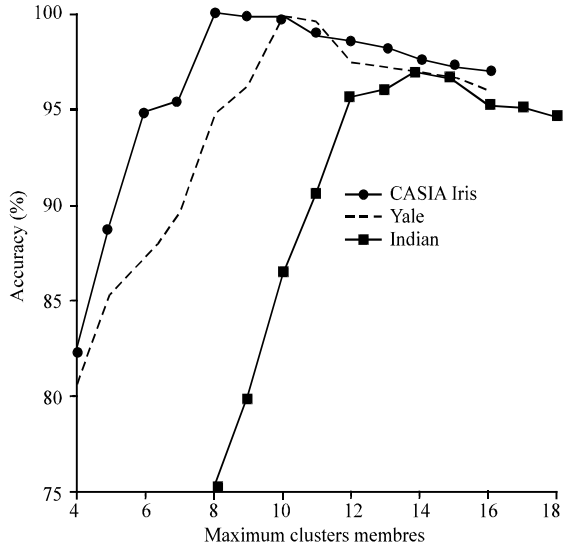

Fig. 12: Impact on accuracy by varying maximum cluster members (T)

no evidential improvement. The consequences of I for the fixed number of hidden neurons and clusters can be scrutinized from Tables 1,4 and 7 and Fig. 12. For CASIA iris, Yale and Indian datasets accuracy go on increasing up to T = 10, T = 8 and T = 14, respectively but later starts degrading. The noteworthy feature of Table 2, 5 and 8 is that classifier based on proposed complex C-TROIKA neuron requires smaller number of hidden neurons than C-MLP to realize the nearly similar accuracy. The corresponding plots are represented in Fig. 3, 6 and 9 for CASIA iris, Yale and Indian datasets, respectively.

CONCLUSION

Finally, we conclude that the proposed C-TROIKA complex neuron model based classifier outperforms C-MLP in terms of recognition accuracy, convergence speed and learning cycles.

REFERENCES

Angelov, P. and A. Sperduti, 2016. Challenges in deep learning. Proceedings of the 24th European Symposium on Artificial Neural Networks Computational Intelligence and Machine Learning, April 27-29, 2016, i6doc Publisher, Bruges, Belgium, ISBN:978-287587027-8, pp: 489-495.

Arivazhagan, S., R.A. Priyadharshini and S. Sowmiya, 2014. Face recognition based on local directional number pattern and ANFIS classifier. Proceedings of the 2014 International Conference on Advanced Communication Control and Computing Technologies (ICACCCT'14), May 8-10, 2014, IEEE, Ramanathapuram,India,ISBN:978-1-4799-3914-5,pp: 1627-1631.

Aslan, M.S., Z. Hailat, T.K. Alafif and X.W. Chen, 2017. Multi-channel multi-model feature learning for face recognition. Pattern Recognit. Lett., 85: 79-83.

Belhumeur, P.N., J.P. Hespanha and D.J. Kriegman, 1997. Eigen faces vs. fisher faces: Recognition using class specific linear projection. IEEE. Trans. Pattern Anal. Mach. Intell., 19: 711-720.

Chen, X., Z. Tang and S. Li, 2005. An modified error function for the complex-value backpropagation. Neural Inf. Process. Lett. Rev., 8: 1-8.

Chen, X.W., M. Aslan, K. Zhang and T. Huang, 2015. Learning multi-channel deep feature representations for face recognition. Proceedings of the 1st International Workshop on Feature Extraction Modern Questions and Challenges, December 7-12, 2015, National Institute of Population Studies (NIPS), Montreal, Canada, pp: 60-71.

Daugman, J.G., 2001. Statistical richness of visual phase information: Update on recognizing persons by iris patterns. Int. J. Comput. Vision, 45: 25-38.

Er, M.J., S. Wu, J. Lu and H.L. Toh, 2002. Face recognition with Radial Basis Function (RBF) neural networks. IEEE. Trans. Neural Netw., 13: 697-710.

Gangal, A.S., P.K. Kalra and D.S. Chauhan, 2007. Performance evaluation of complex valued neural networks using various error functions. Enformatika, 23: 27-32.

Hajari, K., U. Gawande and Y. Golhar, 2016. Neural network approach to iris recognition in noisy environment. Procedia Comput. Sci., 78: 675-682.

Hirose, A., 2006. Complex-Valued Neural Networks. Springer, New York, USA.,.

Hruschka, E.R., R.J. Campello and A.A. Freitas, 2009. A survey of evolutionary algorithms for clustering. IEEE. Trans. Syst. Man Cybern. Part C Appl. Rev., 39: 133-155.

Lukas, S., A.R. Mitra, R.I. Desanti and D. Krisnadi, 2016. Student attendance system in classroom using face recognition technique. Proceedings of the International Conference on Information and Communication Technology Convergence (ICTC), October 19-21, 2016, IEEE, Jeju, South Korea, ISBN:978-1-5090-1326-5, pp: 1032-1035.

Mandic, D.P. and V.S.L. Goh, 2009. Complex Valued Nonlinear Adaptive Filters: Noncircularity, Widely Linear Neural Models. John Wiley & Sons, Hoboken, New Jersey, USA., ISBN:978-0-470-06635-5, Pages: 324.

Nagpal, S., M. Singh, R. Singh and M. Vatsa, 2015. Regularized deep learning for face recognition with weight variations. IEEE. Access, 3: 3010-3018.

Nitta, T., 1997. An extension of the back-propagation algorithm to complex numbers. Neural Netw., 10: 1391-1415.

O'Connor, B., K. Roy, J. Shelton and G. Dozier, 2014. Iris recognition using fuzzy level set and GEFE. Intl. J. Mach. Learn. Comput., 4: 225-231.

Oh, S.K., S.H. Yoo and W. Pedrycz, 2013. Design of face recognition algorithm using PCA-LDA combined for hybrid data pre-processing and polynomial-based RBF neural networks: Design and its application. Expert Syst. Appl., 40: 1451-1466.

Parkhi, O.M., A. Vedaldi and A. Zisserman, 2015. Deep face recognition. Proceedings of the Conference on British Machine Vision (BMVC'15) Vol. 1, September 7-10, 2015, BMVA Press, Swansea, Wales, ISBN:1-901725-53-7, pp: 1-6.

Sheng, W., G. Howells, M. Fairhurst and F. Deravi, 2008. Template-free biometric-key generation by means of fuzzy genetic clustering. IEEE. Trans. Inf. Forensics Secur., 3: 183-191.

Tripathi, B.K. and P.K. Kalra, 2011. On efficient learning machine with root-power mean neuron in complex domain. IEEE. Trans. Neural Netw., 22: 727-738.

Tripathi, B.K., 2017. On the complex domain deep machine learning for face recognition. Appl. Intell., 47: 382-396.

Tseng, L.Y. and S.B. Yang, 2001. A genetic approach to the automatic clustering problem. Pattern Recognit., 34: 415-424.

Wang, Z., M. Jiang, Y. Hu and H. Li, 2012. An incremental learning method based on probabilistic neural networks and adjustable fuzzy clustering for human activity recognition by using wearable sensors. IEEE. Trans. Inf. Technol. Biomed., 16: 691-699.

Yu, J., Q. Cheng and H. Huang, 2004. Analysis of the weighting exponent in the FCM. IEEE. Trans. Syst. Man Cybern. Part B, 34: 634-639.

Zhang, K., W. Zuo, Y. Chen, D. Meng and L. Zhang, 2017. Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising. IEEE. Trans. Image Process., 26: 3142-3155.