Received: October 11, 2017

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

On the Deep Hybrid Computational Model for Face Recognition

Swati Srivastava[1]* **Bipin Kumar Tripathi**[1]

[1] *Department of Computer Science & Engineering,*
Harcourt Butler Technical University, Kanpur, India
* Corresponding author's Email: swati.cs1409@gmail.com

Abstract: This paper aims to develop a recognition system which incorporates a novel fusion from different areas of computational intelligence along with a proposed artificial neuron structure. Our Deep Hybrid Computational Model (DHCM) is a novel aggregation of fuzzy clustering fused with evolutionary searching and a neural network based on a proposed neuron model *TROIKA*. The deep architecture with evolutionary fuzzy clustering followed by hybrid neural network, built upon novel neuron *TROIKA*, yields a powerful tool for face recognition applications. The proposed *TROIKA* neuron is based on nonlinear aggregation functions which enables our hybrid neural classifier to inculcate the benefits of its computational power resulting in superior learning and generalization. The proposed *TROIKA* neuron reduces the complexity of DHCM because very few neurons are sufficient to recognize a subject in database. The experimental results completed on two benchmark face datasets-INDIAN and FERET datasets demonstrated the effectiveness of the proposed model. The performance comparisons revealed the outperformance of (i) *TROIKA* based model over conventional neuron based model (ii) proposed model over other classification methods.

Keywords: *TROIKA* neuron, Integrated fuzzy partitioning (IFP), *TROIKA* based neural classifier (TNC), Deep structure.

1. Introduction

In recent past, various attempts have been made in order to develop deep architectures in the stream of computational intelligence for biometric recognition purpose. The latest researches has scrutinized that a single technique is not as efficient as the fusion of multiple intelligent techniques for recognition, specifically when there are wide variations in the images. Therefore in recent literature, multiple intelligent techniques are combined to augment the performance of biometric based recognition systems. As compared to traditional recognition systems, biometric based recognition systems emerged far better in terms of security and reliability [1]. Researchers have developed various recognition methodologies with different combinations of machine intelligent techniques such as probabilistic neural network with adjustable fuzzy clustering [2], PCA-LDA with polynomial Radial Basis Function (RBF) neural network [3], discrete wavelet transform, discrete cosine transform and RBF based neural network [4]. The objective of all such fusions is to provide better recognition performance. Various state-of-the-art techniques in the context of deep learning have presented the supremacy of deep structures in recent past. Researchers have projected several deep architectures such as denoising convolutional Neural Networks (CNN) [5], deep network architecture consisting of a long sequence of convolutional layers [6], sparse-stacked denoising auto-encoders and deep Boltzman Machines for face recognition [7] to improve the identification performance. In continuation to the above series, we develop a Deep Hybrid Computational Model (DHCM), for recognition purpose, which is a novel fusion of evolutionary fuzzy partitioning and hybrid neural classifier based on proposed neuron model *TROIKA*. We have proposed an artificial neuron model *TROIKA* which incorporates summation and radial basis aggregations along with their product. The proposed network is said to be hybrid since it contains novel *TROIKA* neurons in its hidden layer and output layer is composed of conventional neurons. The sole purpose of this work is to improve the performance of biometric based recognition systems by providing faster convergence, reduced complexity and better prediction accuracy. Proposed DHCM is a novel synergism of unsupervised clustering followed by supervised classification in an innovative manner where evolutionary fuzzy clustering decides the structure of allied hybrid neural network. Also, deep learning is linked with some challenges [8] like computational efficiency and complexity in terms of computational units, topology and learning efficiency. The proposed deep structure cares such issues by providing defined structure of allied networks which clears the complexity of topology. The proposed *TROIKA* neurons are computational efficient which enables the hybrid network to provide quicker convergence and efficient learning.

The paper is organized as follows: Section 2 introduces the proposed Deep Hybrid Computational Model in detail. Experimental results and performance comparisons are presented in Section 3. Section 4 is devoted to discuss the findings of this proposal. Finally, we provide the conclusions and future work in Section 5.

2. Deep hybrid computational model

The deep structure of a proposed model can be vindicated as a ladder of intelligent techniques one followed by another. DHCM consists of two modules: (i) Integrated Fuzzy Partitioning (IFP) (ii) *TROIKA* based neural classifier (TNC). The first module IFP allocates the clusters to the mean feature vectors through fuzzy c-means clustering pursued by evolutionary search technique [9]. The purpose behind using an evolutionary search is to get an optimal distribution of mean feature vectors, as fuzzy clustering standalone gives different distributions in different executions [10]. Let C be the number of clusters, and the number of members that can be allocated to each cluster is referred as maximum cluster members (Ψ). A C x Ψ cluster allocation matrix (∏) is obtained as the outcome of IFP. The C and Ψ are referred to as structural features that are used to define the structure of the proposed allied hybrid neural networks in the second module. Number of clusters decides the number of allied higher order hybrid neural networks and the output neurons in each network will be same in number as Ψ. The proposed TNC module is a hybrid neural classifier which is responsible for learning of pre-classified input patterns. The hybrid network incorporates novel *TROIKA* neurons in its hidden layer while conventional neurons in the output layer. The outcome of the first module is used as the desired output in the second module where the supervised back propagation algorithm with momentum is used for learning purpose. In order to verify the generalization efficiency of the system, the function Maxima of Maxima of outputs of each allied hybrid network is used to determine the resulting class of the test data. Proposed DHCM, as shown in Fig.1, first extracts the desirable features of dataset, then evolutionary fuzzy clustering distribute mean feature vectors into cluster allocation matrix (∏) which is further subject to supervised classification.

2.1 Feature extraction

The extraction of representative and discriminative features is the foremost step of our image recognition system. The Principal Component Analysis (PCA) along with Fisher Linear Discriminant (FLD) is used to extract representative features which also yields dominant discriminant information and optimal subspace representation [11]. It has contributed in better class distinctiveness in classification especially in images with major variations such as pose, facial expression etc. Let ES^T is the transpose of eigen space and A is an array of input images. Let Z be the outcome of PCA which is obtained by projecting input image dataset onto eigenface [20] space as:

$$Z = ES^T . A \tag{1}$$

FLD is applied over $Z = \{Z_1, Z_2, \ldots, Z_n\}$, where n is total number of training images. Maximized ratio of between-class scatter and within-class scatter yields an optimized subspace for classification. The outcome of FLD can be obtained as:

$$F = OS^T \, . \, Z \tag{2}$$

where OS is optimal subspace and can be determined by $\left|\frac{A^T S_b A}{A^T S_w A}\right|$ which gives the solution that is equivalent to the solution of following eigen value problem:

$$S_b \, Z_i = \lambda_i S_w Z_i \tag{3}$$

where $i = 1,2, \ldots, N-1$, S_b is between-class

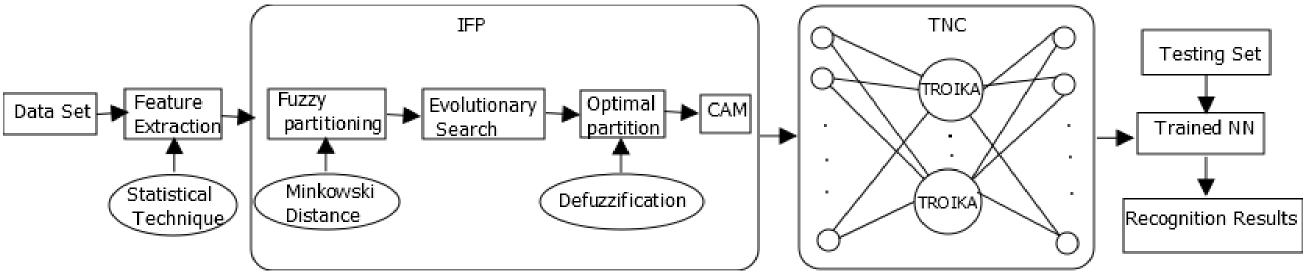

Figure 1. Deep Hybrid Computational Model

scatter, S_w is within-class scatter and N is number of input classes. Here, the eigen vectors corresponding to the first $(N-1)$ largest eigenvalues are selected. S_b and S_w can be defined as:

$$S_b = \sum_{i=1}^{N} S.G.G^T$$

$$\text{where } G = \frac{1}{S}\ \sum_{j=1}^{S} Z_j^i - \frac{1}{n}\ \sum_{j=1}^{n} Z_j \qquad (4)$$

$$S_w = \sum_{i=1}^{N} K.K^T$$

$$\text{where } K = \sum_{j=1}^{S} Z_j - \frac{1}{S}\ \sum_{j=1}^{S}\ Z_j^i \qquad (5)$$

where S is number of sample images in the i^{th} class. Thus, F obtained from Eq. (2) is the outcome of the feature extraction phase. This gives the version of input data which will further be fed into the IFP module of DHCM.

2.2 Integrated fuzzy partitioning

The first module, Integrated Fuzzy Partitioning (IFP), is the integration of fuzzy clustering and evolutionary search which comprises several phases as shown in Fig.1. The outcome of IFP module is cluster allocation matrix ($\prod$) which will further be used in the classification module. This module is divided into different subsections as:

2.2.1. Fuzzy partitioning of mean feature vectors

In this phase, mean feature vectors are partitioned using the concept of fuzzy c-mean clustering. The outcome of feature extraction phase, W$_{\text{FLD}}$, is further compacted into the mean vectors of the input classes which is the actual input to IFP. Let x_{ij} be the i^{th} sample of j^{th} class and q be the number of sample images of j^{th} class involved in the mean computation, where $i = 1,2,\ldots,q$ and $j = 1,2,\ldots,N$. The mean feature vector of j^{th} class can be obtained as:

$$x_{m_j} = \frac{1}{q}\sum_{i=1}^{q} x_{ij} \qquad (6)$$

Let $X = \{x_{m_1}\ , x_{m_2}, \ldots, x_{m_N}\}$ be the mean vectors of N classes. Fuzzy partition matrix $U = [u_{ij}\ ]$ is randomly initialized such that $u_{ij}\ \in [0,1]$ where $i = 1,2,\ldots,C$, $j = 1,2,\ldots N$. Next step to further proceed the fuzzy clustering is to compute cluster centers. Center for i^{th} cluster can be given as:

$$c_i\ = \frac{\sum_{j=1}^{N} u_{ij}{}^{\text{ƒ}} x_{m_j}}{\sum_{j=1}^{N} u_{ij}{}^{\text{ƒ}}} \qquad (7)$$

where parameter ƒ determines the degree of fuzziness and it lies in the range $(1,\infty)$. Performance of the system is greatly influenced by the selection of suitable value of ƒ . Subsequently, the computation of distance between mean vector of j^{th} class and the center of i^{th} cluster is required for the updation of fuzzy partition matrix. Here, the Minkowski distance is used as a measure of distance because of its ability to generate clusters having variable shapes depending on the generalization parameter $\gamma \in [1,\infty)$. Updated U can be obtained as:

$$U_{ij}\ = \left[\frac{\text{ɗ}_{ij}{}^{\frac{-2}{\text{ƒ}-1}}}{\sum_{k=1}^{C} \text{ɗ}_{kj}}\right] \qquad (8)$$

$$\text{where } 1 \le i \le C\ , 1 \le j \le N$$

and $\text{ɗ}_{ij} = \left(\sum_{q=1}^{N-1} \left\|\ x_{m_{jq}} - c_{iq}\right\|^{\gamma}\right)^{\gamma^{-1}}$ is the distance of mean vector of j^{th} class from the center of i^{th} cluster. In order to obtain fuzzy partition matrix, following objective function need to be minimized iteratively:

$$J = \sum_{i=1}^{C}\sum_{j=1}^{N} u_{ij}{}^{F}\ d_{ij} \qquad (9)$$

2.2.2. Evolutionary search to find the best fitted partition

The evolutionary search is considered to find the best partition among the number of different partitions obtained in previous step. Let for r executions, we get r partition matrices represented as $\{U_0\ , U_1\,, \ldots, U_r\}$, which we refer here populations. Fitness function ¥ [12] based evolutionary search technique is used to find the best population among r populations.

$$¥ = \frac{N d_{min}}{\sum_{i=1}^{C}\sum_{j=1}^{N} u_{ij}{}^{\text{f}}\, d} \qquad (10)$$

$$\text{where } d = \sum_{i=1}^{C}\sum_{j=1}^{N} \left\| x_{m_j} - c_i \right\|^2$$

The population with the highest fitness value is the optimal partition matrix 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}.

2.2.3. Defuzzification of optimal partition

The final step of IFP is to defuzzify U_{opt} which results in cluster allocation matrix ($\prod$). To obtain the clusters with fixed number of members, we need to define the maximum cluster member (Ψ). In U_{opt}, we sort all the classes in a cluster according to the degree of membership in descending order. Finally, $\prod$ is obtained by selecting top membership grade elements equal to Ψ. $\prod$ reflects the pre-classification of the training data, for which learning is performed in the neural network. Thus, $\prod$ is used as the desired output in TNC to train the network.

2.3 *TROIKA* based neural classifier

The proposed neuron *TROIKA* based Neural Classifier (TNC), the second module of our proposal, is responsible for two functions: First, for learning of pre-classified training data. Second, for classification of test data. The proposed network incorporates *TROIKA* neurons in its hidden layer and output layer is composed of conventional neurons. Due to the above combination of neurons, TNC is said to be a hybrid network. Proposed neuron *TROIKA* includes summation as well as radial basis aggregation along with compensatory product of these aggregations. The novel neuron embedded with nonlinear functionalities improves the computational power of the proposed network. A three layer higher order hybrid neural network TNC, consisting *TROIKA* neurons in its hidden layer and conventional neurons in output layer is considered for learning and generalization. Such allied hybrid neural network associated with each cluster is ornamented with faster convergence due to computational efficiency of proposed *TROIKA* neuron. In this paper, a neuron with only summation aggregation function is referred to as *conventional neuron* and a neural network built up on these neurons is termed as *(Multi Layer Perceptron) MLP*. Thus, structure of the allied hybrid network is $\{I - H - L\}(C)$ consisting of $I = N - 1$ inputs, H novel *TROIKA* neurons in hidden layer, $L = \Psi$ conventional neurons in output layer; and C is number of clusters. The nonlinear aggregation functions have proved their computational ability in [13]. The beauty of *TROIKA* neuron is its nature of aggregation which takes a broad view of compensatory product of summation and radial basis functions over a set of inputs which enables proposed hybrid classifier to provide faster convergence and better learning efficiency than MLP.

Conventionally, w_{ih} represents the weight from i^{th} neuron to h^{th} neuron. Let input vector be $P = \{p_{1,}p_{2,}\ldots, p_I\}$, $WS_h = \{ws_{1h,}ws_{2h,}\ldots, ws_{Ih}\}$ be the weights from inputs to the summation part of h^{th} novel neuron and $WRB_h = \{wrb_{1h},\ wrb_{2h},\ \ldots,\ wrb_{Ih}\}$ be the weights from inputs to the radial basis part of h^{th}novel neuron. Let $W_{0h} = \{w_{01}, w_{02}, \ldots, w_{0H}\}$ be the bias weight and ph_0 be the bias input for H *TROIKA* neurons in the hidden layer. Let $W_l = \{w_{1l}, w_{2l}, \ldots, w_{Hl}\}$ be the weight vector of hidden neurons to l^{th} output neuron. Let $W_{0l} = \{w_{01}, w_{02}, \ldots, w_{0L}\}$ be the bias weight vector and pl_0 be the bias input for L conventional neurons in the output layer. The net input of h^{th} *TROIKA* neuron can be computed through generalized product as:

$$V_h = vs_h + vrb_h + vs_h vrb_h + w_{0h} ph_0 \qquad (11)$$

where $vs_h$and vrb_h are the potentials for summation and radial basis parts of h^{th} *TROIKA* neuron and can be determined as:

$$vs_h = Q_h \sum_{i=1}^{I} p_i ws_{ih} \qquad (12)$$

$$vrb_h = R_h e^{-\sum_{i=1}^{I} \| p_i - wrb_{ih} \|^2} \qquad (13)$$

where Q_h and R_h indicates the contribution of summation and radial basis parts of h^{th}novel neuron respectively. The net output of h^{th} *TROIKA* neuron is given as:

$$Y_h = f(V_h) \qquad (14)$$

where f is an activation function whose selection depends on the nature of the application. The net potential and output of l^{th} conventional neuron is respectively given by :

$$V_l = \sum_{h=1}^{H} w_{hl} Y_h + w_{0l}\, pl_0 \qquad (15)$$

$$\text{and } \; Y_l = f(V_l) \qquad (16)$$

Let Y_d be the desired output, then the error at l^{th} neuron ($e_l = Y_{dl} - Y_l$) yields the mean squared error (MSE)

$$E = \frac{1}{2} \sum_{l=1}^{L} {e_l}^2 \qquad (17)$$

2.3.1. Learning rules with *TROIKA* based network

The back-propagation learning of proposed neural classifier is based on gradient descent approach. Addition of momentum term allows a network to slide through local minima in the error surface, if occur. The minimization of E, drives weight update equation as:

$$w_{new} = w_{old} + \Delta w^{(t)} + \alpha\, \Delta w^{(t-1)} \qquad (18)$$

where $\Delta w = -\eta \nabla_w E$

$$\Delta w = -\eta \frac{\partial E}{\partial w} \qquad (19)$$

where η is the learning rate and α is the momentum factor whose values ranges in between 0 and 1. Let f' be the derivative of the activation function f. Weight update equation for l^{th} neuron at output layer can be obtained as:

$$-\frac{\partial E}{\partial w_{hl}} = e_l f'(V_l) \frac{\partial V_l}{\partial w_{hl}} \qquad (20)$$

$$\Delta w_{hl} = \eta \; e_l f'(V_l)\, Y_h \qquad (21)$$

$$\text{Similarly, } \; \Delta w_{0l} = \eta \; e_l f'(V_l)\, pl_0 \qquad (22)$$

Weight update in *TROIKA* neuron for $w = w_{ih}$ can be generalized as:

$$-\frac{\partial E}{\partial w_{ih}} = \frac{\partial V_h}{\partial w_{ih}} f'(V_h) \sum_{l=1}^{L} e_l f'(V_l) w_{hl} \qquad (23)$$

Weight update equation for summation part of *TROIKA* neuron can be obtained by rewriting Eq. (23) as,

$$-\frac{\partial E}{\partial ws_{ih}} = \frac{\partial V_h}{\partial ws_{ih}} B_h \qquad (24)$$

where $B_h = f'(V_h) \sum_{l=1}^{L} e_l\, f'(V_l) w_{hl}$
and substituting Eq. (24) in Eq. (19),

$$\Delta ws_{ih} = \eta\, p_i\, Q_h (1 + vrb_h) B_h \qquad (25)$$

Similarly, weight update equation for radial basis part of *TROIKA* neuron is given as:

$$\Delta wrb_{ih} = 2\eta \frac{vrb_h}{R_h} (\, p_i - wrb_{ih}\,) B_h R_h\, (1 + vs_h\,) \qquad (26)$$

The update equations for compensatory weights associated with summation and radial basis contributions of *TROIKA* neuron can be given as:

$$\Delta Q_h = \eta \frac{vs_h}{Q_h} \,(1 + vrb_h\,) B_h \qquad (27)$$

$$\text{and } \; \Delta R_h = \eta \frac{vrb_h}{R_h} \,(1 + vs_h\,) B_h \qquad (28)$$

$$\text{Bias Update, } \quad \Delta w_{0h} = \eta\, B_h\, ph_0 \qquad (29)$$

Update the weights recursively until the proposed classifier is trained according to $\prod$ obtained in the previous module. The trained neural networks are further used to classify the test data.

2.3.2. Merit based decision

Once we get trained neural networks, testing can be performed by using the function *Maxima of Maxima* of the outputs of each allied neural network. For every input test pattern, we have obtained the $C\ x\ \Psi$ output matrix. Let $O_k(NN_i)$ be the maximum output of i^{th} allied NN for k^{th} testing pattern. The resulting class of a test pattern can be identified by finding the cluster corresponding to $MAX_{i=1}^{C}(O_k(NN_i))$.

3. Performance evaluation

The performance of the proposed model has been assessed on two benchmark facial datasets: INDIAN face dataset [14] and FERET face dataset [15]. The evaluation of our proposal is based on different parameters such as number of hidden neurons (*TROIKA*), number of clusters (C), maximum cluster members (Ψ) , False Acceptance Rate (FAR) and False Rejection Rate (FRR) [1].The low values of FAR and FRR indicates the efficacy of a system. It is worth mentioning here that when we say TNC, it means the network which has *TROIKA* neurons in its hidden layer and conventional neurons in its output layer whereas MLP reflects the network which has conventional neurons both in its hidden as well as output layers. Thus, the two aforementioned networks can be differentiated by their hidden layers. The

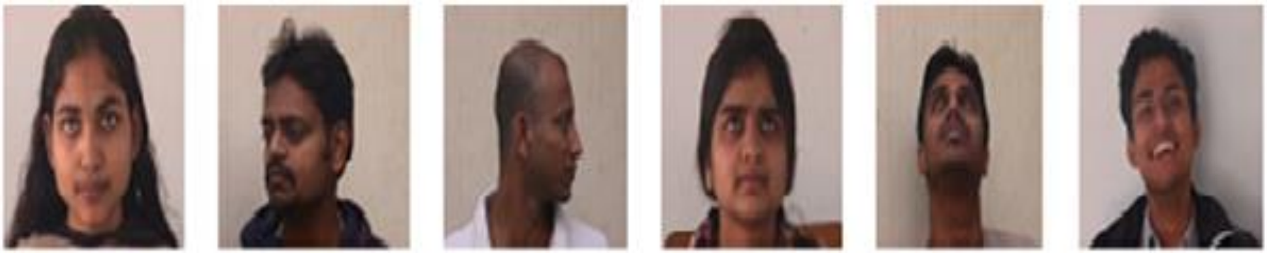

Figure.2 Sample images from INDIAN Face Dataset

Table 1. Recognition accuracy for INDIAN Face Dataset

***TROIKA* Neurons (H)**	***C***	***Ψ***	**MSE**	**Accuracy (%)**	
				Training set	**Testing set**
4	8	12	.006025	96.2	82.2
		14	.007443	97.2	85.4
		16	.009159	97.0	85.1
	10	12	.005421	97.2	88.4
		14	.007516	98.2	92.3
		16	.010759	98.0	92.1
	12	12	.005848	96.9	88.2
		14	.008524	98.0	92.0
		16	.009957	98.0	91.8
6	8	12	.004284	98.6	84.3
		14	.005867	99.0	86.8
		16	.008238	98.9	86.2
	10	12	.004765	99.8	93.6
		14	**.006527**	**99.9**	**96.5**
		16	.009235	99.9	96.0
	12	12	.003807	99.8	93.0
		14	.005041	99.9	96.3
		16	.011896	99.8	95.9
8	8	12	.002574	98.6	84.0
		14	.003666	98.8	86.2
		16	.009114	99.0	86.0
	10	12	.003474	99.8	93.2
		14	.004687	99.8	96.0
		16	.008745	99.8	95.6
	12	12	.003019	99.6	92.8
		14	.003951	99.8	95.7
		16	.007896	99.8	95.0

performance comparisons has been done in two ways: First; we compare the proposed model on the basis of variations in second module; TNC and MLP. The accuracy results of the proposed classifier TNC are compared with MLP. The basis of comparison is the neuron type used in hidden layer of both the neural networks and therefore the number of hidden neurons is the actual parameter of comparison. Second; we have compared the accuracy results of proposed model with other models.

3.1 Evaluation on INDIAN face dataset

This dataset contains the face images of female and male subjects with different orientations and each subject consists of 10 images. The variations in images of dataset can be seen from Fig. 2. We have taken a subset of 500 images corresponding to 50 subjects to evaluate the performance of the proposed

Table 2. Comparison based on hidden neuron type used in MLP and TNC for INDIAN face dataset

Classifier	H	C	Ψ	MSE	Accuracy (%)	
					Training Set	Testing Set
MLP	10	8	14	.007480	96.8	85.5
			16	.009406	96.6	85.0
		10	14	.008294	98.2	92.5
			16	.010479	98.2	92.1
		12	14	.008226	98.8	91.7
			16	.009994	98.8	91.2
TNC	4	8	14	.007443	97.2	85.4
			16	.009159	97.0	85.1
		10	14	.007516	98.2	92.3
			16	.010759	98.0	92.1
		12	14	.008524	98.0	92.0
			16	.009957	98.0	91.8

Figure.3 FERET face dataset sample images

Table 3. Comparison with other methods for INDIAN face dataset

Methods	Recognition accuracy (%)
ANFIS Classifier [16]	90.47
OCON Classifier [17]	95.0
DHCM [Proposed]	96.5

system. Here 4 images of each subject were used for training and remaining 6 images of each subject were used for testing.

Table 1 unveils the comparative analysis for accuracy with the variation in hidden neurons, clusters and maximum cluster members. The effect of variation of one parameter on the recognition accuracy can be observed by keeping the other two parameters constant. Table 2 brings out the fact that proposed TNC outperforms MLP as very few proposed hidden neurons are able to achieve the approximately same results while adequately large number of conventional neurons are required with MLP. For the same values of C and Ψ, one can examine the aptitude of hidden neurons for MLP and TNC. Table 3 summarized a comparison of proposed model with other models which attested the better training and testing accuracy of proposed DHCM. Table 7 demonstrates that FAR and FRR for TNC is recorded as 0.8% and 3.5% respectively that are less than FAR and FRR for MLP for same set of other parameters, which again designates the improved performance of proposed *TROIKA* neuron. The best obtained accuracy for INDIAN dataset is 96.5% when number of $TROIKA = 6, C = 10$ and $\Psi = 14$.

3.2 Evaluation on FERET face dataset

FERET is a well-known standard face database for research which contains the face images of 1208 persons. Fig. 3 shows the variations in images of dataset. A subset of 100 subjects have been considered to evaluate the performance of DHCM. From each subject, 4 images were considered for training and 6 images were engaged in testing.

Table 4 discloses the recognition accuracy of FERET face dataset where the effect of variation of hidden neurons can be seen apparently keeping other parameters constant. The superiority of the proposed neuron *TROIKA* is again witnessed from Table 5 as TNC requires very few neurons in hidden layer as compared to MLP, to realize the similar accuracy.

Table 6 compares the classification accuracy of our proposal with other models which encourages the better performance of the proposed deep learning system. Further, with same number of other parameters, Table 7 shows the 2.6% FAR and 5.6%

Table 4. Recognition Accuracy for FERET face dataset

***TROIKA* Neurons (H)**	***C***	***Ψ***	**MSE**	**Accuracy(%)**	
				Training set	**Testing set**
6	8	12	.007995	97.8	86.2
		14	.008108	98.0	90.5
		16	.009865	98.0	90.0
	10	12	.007954	98.6	89.9
		14	.008622	98.8	92.4
		16	.010932	98.8	92.1
	12	12	.008289	98.4	89.5
		14	.009312	98.8	92.2
		16	.009682	98.8	91.5
8	8	12	.008316	99.0	89.9
		14	.008564	99.5	92.7
		16	.009205	99.5	92.3
	10	12	.007635	99.6	91.8
		14	**.008904**	**99.6**	**94.4**
		16	.009237	99.6	94.2
	12	12	.008856	99.4	91.5
		14	.009506	99.5	94.0
		16	.009989	99.5	93.8
10	8	12	.008552	99.0	89.5
		14	.008962	99.5	92.2
		16	.009632	99.5	92.0
	10	12	.007915	99.6	91.4
		14	.008026	99.6	94.0
		16	.009892	99.6	93.6
	12	12	.007999	99.4	90.9
		14	.009128	99.5	93.6
		16	.0097531	99.5	93.1

Table 5. Comparison based on hidden neuron type used in MLP and TNC for FERET face dataset

Classifier	***H***	***C***	***Ψ***	**MSE**	**Accuracy (%)**	
					Training set	**Testing set**
MLP	15	8	14	.008452	96.5	90.8
			16	.009436	96.4	90.2
		10	14	.008574	98.8	92.6
			16	.010478	98.6	92.0
		12	14	.008223	98.8	91.9
			16	.009998	98.8	91.2
TNC	6	8	14	.008108	98.0	90.5
			16	.009865	98.0	90.0
		10	14	.008622	98.8	92.4
			16	.010932	98.8	92.1
		12	14	.009312	98.8	92.2
			16	.009682	98.8	91.5

FRR for the proposed model while for MLP based model the FAR is 3.0% and FRR is 6.8%. The lower values of FAR and FRR for the proposed model again demonstrated the efficacy of *TROIKA* neuron.

For proposed hybrid system based with *TROIKA* =8, C=10 and Ψ=14 we attain best recognition accuracy of 94.4% in FERET dataset.

Table 6. Comparison with other methods for FERET face dataset

Methods	**Recognition Accuracy (%)**
Linear Regression Classification Model [18]	87.11(EP1), 84.50 (EP2)
Dual Cross Patterns face image descriptor [19]	92.80
DHCM [Proposed]	94.4

Table 7. FAR and FRR for INDIAN and FERET datasets for both MLP and TNC based models

		INDIAN	**FERET**
MLP	FAR (%)	1.2	3.0
	FRR (%)	4.4	6.8
	Accuracy (%)	96.0	93.2
TNC	FAR (%)	0.8	2.6
	FRR (%)	3.5	5.6
	Accuracy (%)	96.5	94.4

4. Discussion

In previous section, first we have performed experiments on two famous benchmark facial datasets for the proposed DHCM model where we have used proposed TNC as a classifier in second module. From Tables 1 and 4, the effect of varying the number of hidden neurons (*TROIKA*), clusters (C) and maximum cluster members (Ψ) on the accuracy of the system can be observed. Fig. 4 better presents their variations for prospective readers. In our experiments, for INDIAN dataset, best accuracy is recorded when *TROIKA* is 6 whereas for FERET dataset best results are obtained at *TROIKA* equal to 8. For both datasets, significant improvement in accuracy is observed up to certain number of hidden neurons for constant Ψ and clusters, after that no prominent improvement is recorded. For fixed number of hidden neurons and Ψ, one can monitor the effect of number of clusters. For INDIAN and FERET datasets, we observe that recognition rate enhances up to $C = 10$, after which there is no evidential improvement. The upshots of Ψ can be inferenced by keeping number of hidden neurons and clusters constant. For INDIAN and FERET datasets, accuracy go on increasing up to $\Psi = 14$ but later starts degrading.

Next, we have also performed experiments on same datasets to evaluate our proposed model using MLP as a classifier in second module. All experiments have performed under same conditions except the number of hidden neurons for both the classifiers viz. MLP and TNC. Thus, we have compared TNC with MLP on the basis of hidden neurons. From Tables 2 and 5, it is clearly visible that very few *TROIKA* neurons are able to achieve the estimated results as compared to conventional neurons which apparently demonstrated that proposed *TROIKA* neurons are computationally efficient than conventional neurons. The ensemble of novel *TROIKA* neurons in proposed TNC enables the proposed model DHCM to provide improved learning speed and better recognition accuracy with reduced complexity than MLP. At last, we have compared our proposal with existing methodologies on the basis of best accuracies obtained for INDIAN and FERET datasets which attested the effectiveness of the proposed recognition model, DHCM. Our model provides faster convergence with reduced computational complexity because very few *TROIKA* neurons are sufficient as compared to conventional neurons for recognition purpose which leads to the reduced computations.

5. Conclusion

In this paper, we have proposed an artificial neuron model, *TROIKA*, based on nonlinear aggregations. Next, we have proposed *TROIKA* based hybrid neural classifier (TNC) which consists of novel neurons in its hidden layer and conventional neurons in its output layer. Finally, we presented a deep computational model, DHCM, based on proposed TNC classifier. All experiments and comparisons recorded in this paper evidenced that our deep computational intelligent system attested the beauty of proposed *TROIKA* neuron and the novel synergism of evolutionary fuzzy clustering with TNC. The proposed *TROIKA* neuron based deep hybrid model has thoroughly demonstrated its outperformance over conventional neuron based model. Also, the accuracy comparisons in previous section proves the effectiveness of proposed model over other methodologies. In future, to further improve the recognition accuracy, we can use the complex back-propagation algorithm for learning instead of real back-propagation.

References

[1] S. Srivastava and S. Agarwal, "Offline Signature Verification using Grid based Feature Extraction", In: *Proc. of International Conf. on Computer & Communication Technology*, Allahabad, pp.185-190, 2011.

[2] Z. Wang, M. Jiang, Y. Hu, and H. Li, "An incremental learning method based on probabilistic neural networks and adjustable fuzzy clustering for human activity recognition by using wearable sensors", *IEEE Transactions On Information Technology In Biomedicine*, Vol. 16, No.4, pp.691–699, 2012.

[3] S. K. Oh, S. H. Yoo, and W. Pedrycz, "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 Systems with Applications*, Elsevier, Vol.40, No. 5, pp.1451–1466, 2013.

[4] S. Lukas, A. R. Mitra, R. I. Desanti, and D. Krisnadi, "Student Attendance System in Classroom using Face Recognition Technique", In: *Proc. of International Conf. on Information and Communication Technology Convergence*, pp. 1032-1035 , 2016.

[5] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising", *IEEE Transactions on Image Processing*, Vol.26, No.7, pp. 3142-3155, 2017.

[6] O. M. Parkhi, A. Vedaldi. and A. Zisserman," Deep Face Recognition", *British Machine Vision Conference*, Vol. 1, No. 3, p. 6 , 2015.

[7] S. Nagpal, M. Singh, R. Singh, and M. Vatsa, "Regularized Deep Learning for Face Recognition With Weight Variations", *IEEE Access 3*, pp.3010-3018, 2015.

[8] P. Angelov and A. Sperduti, "Challenges in Deep Learning", In: *Proc. of the 24th European symposium on Artificial Neural Networks*, pp. 489-495, 2016.

[9] B. Yuan, G. J. Klir, and J. F. Stone, "Evolutionary fuzzy c-means clustering", In: *Proc. of International Conf. on Fuzzy System*, pp. 2221–2226, 1995.

[10] E. R. Hruschka, R. J. G. B. Campello, A. A. Freitas, and A. C. P. L. F. de Carvelho, "A survey of evolutionary algorithms for clustering", *IEEE Transactions on System Man And Cybernetics-Part C: Applications and Reviews* , Vol.39, No.2, pp.133–155, 2009.

[11] M. J. Er, W. Shiquian, L. Juwei, and L. T. Hock, "Face recognition with radial basis function neural networks", *IEEE Transactions on Neural Network*, Vol.13, No.2, pp. 697–709, 2002.

[12] W. Sheng, G. Howells, M. Fairhurst, and F. Deravi, "Template free biometric key generation by means of fuzzy genetic clustering", *IEEE Transactions In Information Forensic and Security*, Vol.3, No.2, pp.183-191, 2008.

[13] B. K. Tripathi and P. K. Kalra, "On Efficient learning Machine with Root-Power Mean Neuron in complex ssdomain", *IEEE Transactions on Neural Networks*, Vol.22, No.5, pp.727-738, 2011.

[14] http://viswww.cs.umass.edu~vidit/IndianFace Database.

[15] Feret face database,"https://www.nist.gov/itl/iad/ image-group/color-feret-database".

[16] S. Arivazhagan , R A. Priyadharshini, and S. Sowmiya, "Face Recognition based on Local Directional Number Pattern and ANFIS Classifier", In: *Proc. of International Conf. on Advanced Communication Control and Computing Technologies* , pp. 1627-1631, 2014.

[17] B.K. Tripathi, "On the complex domain deep machine learning for face recognition", *Applied Intelligence*, Springer, pp.1-15, 2017.

[18] I. Naseem, R. Togneri, and M. Bennamoun, "Linear Regression for Face Recognition", *IEEE Transactions On Pattern Analysis And Machine Intelligence*, Vol. 32, No. 11, pp. 2016-2112, 2010.

[19] C. Ding, J. Choi, D. Tao, and L. S. Davis, "Multi-Directional Multi-Level Dual-Cross Patterns for Robust Face Recognition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 38, No.3, pp.518-531, 2016.

[20] P. Belhumeur, J. Hespanha, and D. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19, No.7, pp.711-720, 1997.