Design of N-Way DBN Classifier and Auto Encoder for Facial Similarity

Humera Tariq
University of Karachi, Pakistan
Main University Road, Karachi 75270, Pakistan
Tel.: +92 21 99261300
humera@uok.edu.pk

Zainab Parveen
University of Karachi, Pakistan
Main University Road, Karachi 75270, Pakistan
Tel.: +92 21 99261300
zinnia_zp17@yahoo.com

Abstract

Recognizing a face is a routine task for human perception system, whereas building a similar computational model for Face recognition with 100% accuracy is quite challenging. This work is rich in terms of theory, literature and experimentation with a novel approach towards face similarity technology. The paper concentrates on design of an N-Way DBN Classifier and Similarity Metric DBN Auto-encoder for recognizing facial similarity (or Face Verification). PCA feature extraction process is combined with DBN to enhance the performance of the two proposed models. The models work under both Image Unrestricted and Image Restricted settings on the small LFW dataset. Many improvements and refinements needed to improve the models. With advancement in feature extraction stage, the proposed approaches might achieve competitive verification performance.

Keywords: Face Recognition, Deep Belief Network, Restricted Boltzmann Machine (RBM), Auto Encoder

1. Introduction

The work in this paper explores Deep Belief Network (DBN) for classification of facial images as a matching or non-matching pair based on similarity score between two faces. Classifying a pair of facial images as matching or non-matching is analogous to Face Verification. Thus the problem states that : Given a pair of aligned facial images $P = \{F_1, F_2\}$, the DBN based binary classifier assigns a label ℓ to the pair P. The assigned binary labels $\mathcal{L} = \{1, 0\}$ states whether the given pair is similar $(\ell = 1)$ or dissimilar $(\ell = 0)$.

Recent era came up with different flavors of Artificial Neural Networks (ANN) with Deep Architectures to attain state of the art results for face recognition (Pandaya et.al, 2013)(Zhao et.al, 2003)(El-Sayed, M.A. & Hamed,K.,2015). Such flavors include Deep Neural Networks (DNN), Deep Belief Networks (DBN), Convolutional Neural Networks (CNN), Deep Boltzmann Machines (DBM), etc. DBN learning overcomes the problem of difficult optimization task of deep architectures with success, leaving behind the state-of-the-art in certain areas (Youshua 2009). Deep Learning or Hierarchical Learning uses one of a set of algorithms that strives to model high level abstractions in data by using architectures composed of multiple non-linear transformations. The aim of deep learning is to take low-level inputs (feature vectors) and transform it into higher and higher abstract representations through the hierarchy of layers. These Deep architectures have many layers and parameters that must be learnt. Huge number of layers makes the training, time consuming and training becomes trapped at local minima which leads to unaccepTable results (Hinton, 2009). DBNs have many applications like image recognition and classification, speech recognition, natural language processing, data reconstruction, noise reduction, video sequences and motion capture data (Bedre et.al, 2012). We classify and reconstruct images using *DeeBNet Toolbox* which is a remarkable work of *Mohammad Ali Keyvanrad and Homayounpour MM*. The toolbox is open source and freely available on the internet. The foremost step in any Face Recognition or Face Classification System is to search for the presence of face(s) in any given image. Prior to face detection, pre-processing is done to reduce noise and variation in illumination, pose and expressions. Enhancement, resizing and handling image orientations are usual preprocessing steps and they greatly affect the face detection process. Followed by Face detection, there comes Feature Extraction which can be broadly divided into three categories: (1) line, edges and curves based for e.g. Line Edge Map (LEM); (2) template matching of prominent features such as eyes but this approach is ineffective in handling large variations in appearance of features; (3) geometrical constraints based facial feature extraction. This approach is effective in dealing with variations in image intensity and feature shape. For example, Active Shape Model (ASM), Flexible Appearance Model (FAM), Active Appearance Model (AAM). The extracted features are now examined or compared with the database or gallery of previously stored facial images. Comparison is done to find a match or further classification is done to name/label the subpopulation to which new observation belongs. It either identifies or verifies a query image. The two main task comprising Face Recognition are *Identification* and *Verification* whose comparison is shown in Table 1. Recognition and classification are so much interrelated that defining them precisely is almost impossible. However, Recognition is the process of identifying and locating things (e.g. face) in an image. Classification is the process of categorizing the given data (images) into its subcategory or subpopulation.

Table 1. Face Identification vs. Face Authentication

Face identification	Face verification/ authentication
1: N match that compares a query face image or probe against all image templates in a face database (gallery).	1:1 match that compares a labeled face image against a valid template to check for the label being claimed.
Input: unlabeled and unknown	**Input:** labeled face
Output: Determines identity or class.	**Output:** Yes/NO, confirms or rejects the claimed label or identity of input face.
It can be viewed as an ***N***-ways face classification which is a multiple class classification problem.	It is a binary classification problem which determines similarity or dissimilarity between two face images.

Rest of the paper is organized into following sections: Section 2 reviews the rigorous and rich literature of face recognition and summarizes its detail in tabular format. In Section 3 we briefly describe the theory of Deep Belief network and try to explain its concepts in simpler way as its mathematics foundations are difficult to absorb. Followed by face recognition; we describe the related work in Section 4, which is our inspiration to work with DBN for face recognition. In Section 5 we explain the usage of *DeeBNet (**Dee**p **B**elief **Net**work)* toolbox. Section 6 is all about experimentation with different datasets. Section 7 discuss Quantitative analysis and discussion of results. Finally, Conclusion and future work is given in Section 8.

2. Face Recognition Techniques

There are different approaches to input a *2D* image as input to the Face Recognition System (FRS): *Holistic matching methods* use the whole face region as input to recognition system to report a match. For example, Eigen faces. *Feature-based methods* use extracted facial features, their location and statistics to recognize a match. For Example, Elastic Bunch Graph Matching (EBGM) or Dynamic Link Matching (DLA). As human perception system recognize face using features as well as whole face, *hybrid matching methods* use extracted features from whole image to mimic human behavior. The categorization of some well-known Face Recognition Techniques (FRTs) based on these approaches is shown in Figure 1.

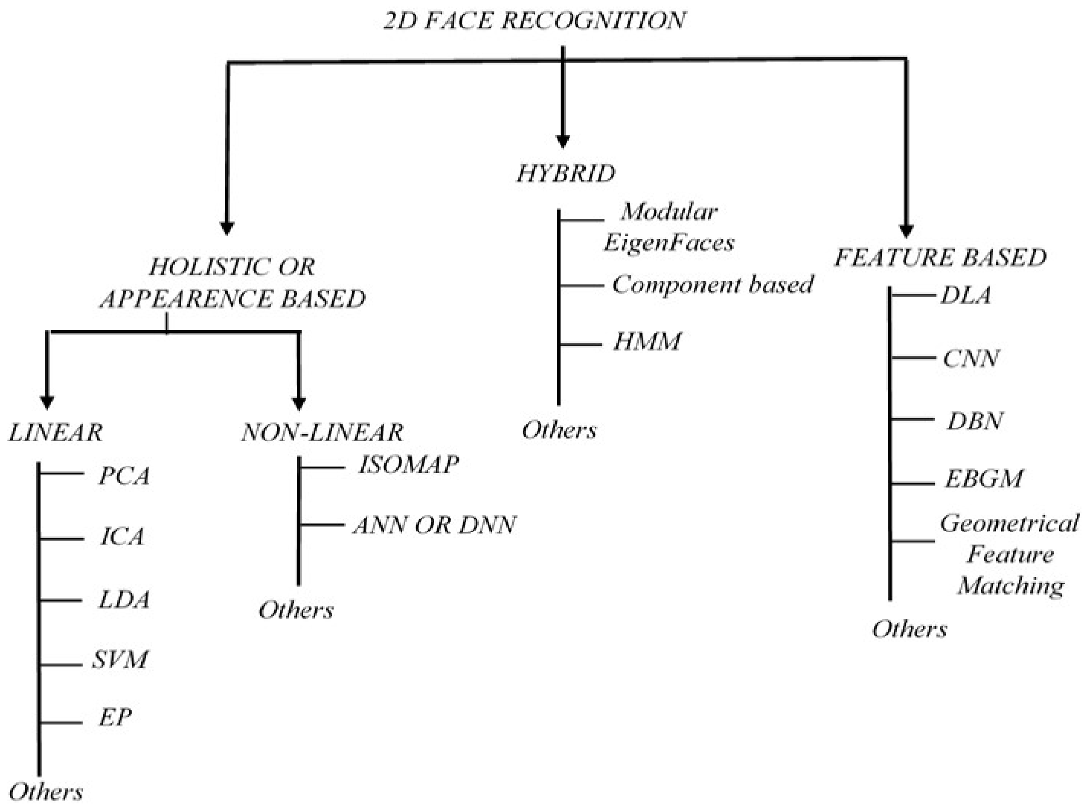


Figure 1. Overview of 2D image Recognition and classification approaches

Face recognition is an active research area since 1990. The popular techniques since 1990 till 2000 includes: Principal Component Analysis (PCA), Independent Component Analysis (ICA), Linear Discriminant Analysis (LDA), Artificial Neural Network (ANN), Gabor wavelets and Support Vector Machines (SVM). *Principal Component Analysis (PCA)* is a well-known algorithm for dimension reduction and feature selection. It is also known as *Karhunen-Loeve Transform (KLT)*. It can be used for reconstruction of human face as well as for recognition. It performs best for frontal face images but is sensitive to face orientation, position and illumination. ICA find components from multidimensional statistical data and is a generalized form of PCA which outperforms PCA and replaces it in many applications. *Linear Discriminant Analysis (LDA)* is a generalized form of *Fisher's Linear Discriminant (FLD)* which finds a linear combination of discriminatory features by considering within class and between class scatter as the relevant information for pattern recognition. It needs large sample size for good generalization. *Artificial Neural Networks (ANN)* provides an attractive method for face recognition because of its non-linearity and its capability to capture complex information of face patterns through training. But it is computationally expensive when training large datasets. It provides partial invariance to translation, rotation, scale and deformation. *Gabor wavelets* are invented by Dennis Gabor using complex functions which serves as a basis for Fourier transform in information theory applications. Gabor wavelets have many applications in image analysis and feature extraction due to its multi-resolution and multi-orientation properties. The frontal face image is convolved with Gabor wavelets and the resulting image are processed for recognition process. It has higher recognition rate and classification accuracy even for low dimension images. It may be considered as the best method for face recognition process. *Support Vector Machine (SVM)* are supervised learning models that can be applied to classification and regression analysis. The SVM classifier has a better generalization performance over traditional neural network. But cannot be used when feature vectors defining samples have missing entries. It is a great method for classification but requires large memory and computation time when dealing with large datasets. The summarized analysis of Holistic Matching Methods and Feature based methods is provided in Table 2 and Table 3 respectively. There are methods which can be used in either way i.e. as holistic or as feature based. This property is attained by Genetic Algorithms. Liu and Welscher propose Evolution Pursuit (EP) technique which uses a combination of PCA and GA to process face image in low-dimensional subspace. It provides a global searching method which provides better results than normal PCA and template matching method but to search for face of interest, a lot of data needs to be processed which makes it expensive. Since 2000, there has been many breakthroughs and advancements in the field of Face Recognition (Ian Good et.al, 2016). Most of the work is done with SVMs and Deep Architectures like DBN, CNN, DNN, etc. The noTable publications timeline is listed in Table 4 and Table 5 respectively.

3. Theory

Before we dive in to understand DBN, it is important to first understand building blocks of DBN which includes Restricted Boltzmann Machines (RBM) and Sigmoid Belief Network (SBN) [1]. RBM was first introduced under the name of **Harmonium** and is particularly an *Energy-based undirected probabilistic graphical model* which contains a layer of stochastic visible units or input variables (say $\boldsymbol{v}$) and a single layer of stochastic hidden units or latent variables (say $\boldsymbol{h}$). Energy-based probabilistic models without hidden unit define the probability distribution as follows:

$$P(x) = \frac{e^{-Energy(x)}}{\sum_x e^{-Energy\,(x)}}$$

Where $\sum_x e^{-Energy\,(x)}$ is the normalizing factor $\boldsymbol{Z}$, and is known as the **partition function** (Youshua, 2009) . $\boldsymbol{Z}$ is the value which ensures that probability distribution sums or integrates to 1. The reason for the exponent is that the undirected models are based on the assumption that all values for $p(x) > 0$. Hence, Energy-based models enforce this condition as exponent is positive for all values (LeCun, 2006).

Table 2. Summary of Holistic Face Matching Methods till Year 2000

Algo-rithm	Author & Year	Technique	Explanation	Pros	Cons
PCA	(Turk and Pentland,1991), (Kirby and Sirovich, 1990)	Eigen Face or Karhunen-Loeve Expansion	Any face image could be reconstructed with a small collection of weights called Eigen picture or Eigen face.	Good for dimension reduction. Effective for frontal face images.	Sensitive to variation in head pose and illumination.
FLD/LD A	(Belhumeur, 1997), (Zhao, 1999)	Fisher Face/ Subspace LDA	Similar to Eigen face method except that it uses combination of PCA and LDA	Works better for variation in light	Requires large sample size for training
ICA	(Bartlett,1998)	ICA based feature Analysis	Used for finding component from multi-dimensional data	Better than PCA for variation in pose, lightening and expression	Does not provide enough independence to variations in PIE.

Table 3. Summary of Feature Based Face Recognition Methods till Year 2000

Algo-rithm	Author & Year	Technique	Explanation	Pros	Cons
ANN	(Lin, 1997)	Probability Decision Based Neural Network (PDBNN)	Has merits of both NN and statistical approaches.	Beneficial for hardware execution such as distributed computing. It has high recognition	Computationally expensive for training large datasets.

				rate.	
		CNN	Convolutional NN are composed of layers of 2 types: convolutional layer provides local connections between neurons and subsampling layers which organize hierarchical mapping of the image.	It provides rotational invariance.	Computationally expensive when training large datasets. Many numbers of layers with fully-connected NN makes optimization almost impossible when parameters are set randomly[15].
	(Lee, Grosse and Ranganath n.d.), (Liu, Zhong and Liu 2011)	DBN	DBN is a generative model which mixes directed and undirected probabilistic connections between variables.	DBN tackles the problem of difficult optimization tasks in Deep models with notable success. It can be used with supervised or unsupervised training.	Training a large dataset with huge number of parameters is time consuming.
Graph Matching (Extension of ANN and Gabor Wavelet)	(M.Lades, 1993), (Okada,1998)	Dynamic Link Architecture	Faces are represented as graphs and edges labeled with 2D distance vector.	Best of all in terms of rotation invariance.	Complex and computationally expensive.
	(Wiskott, 1997)	Elastic Bunch Graph Matching (EBGM)	Face is represented by a data structure called bunch graph and Gabor wavelets are used to produce local features of face image.	Can deal with missing features,	Sensitive to lighting conditions
Geometrical Feature Matching	(kanade,1973), (Kelly,1970)		Works on the feature vector of geometrical features from the face image (e.g. Position and size of eyes, nose, outline, etc.)	GFM based on precisely measured distance is useful for finding possible matches in large database.	Too many parameters are required for matching.
SVM	(Guo Li, Kapluk Chan, 2000)		SVM is a supervised learning model which can be used for classification or regression analysis.	It provides better induction results than ANN for supervised training.	It requires a set of complete features for classification. It cannot deal with missing entries. Requires large memory and computation time.

Table 4. Summary of Face Recognition Literature from Year 2000 till 2010

Year	Authors	Title	Explanation
2000	(Guodong Guo, Li, S.Z., Kapluk Chan)	FR using SVM	SVM was first time used for FR and outperformed all other techniques. SVM was used as holistic method
2001	(Heisele, Ho, Poggio, T)	FR using SVM: Global vs. Component Based Approach	SVM was used as Feature Based Method which provided much better results than the holistic method.
2002	(Dihua Xi; Podolak, I.T., Seong-Whan Lee)	Facial Component Extraction and FR using SVM	SVM was used on dataset with variation in pose and SVM proved to be a fast algorithm.
2003	(Guang Dai, Changle Zhou)	FR using SVM with Robust Feature	A combination of Gabor Wavelets, KPCA was used and SVM was used for classification. SVM provided results with high accuracy.
2004	(Safari, Harandi, Araabi)	A SVM based method for FR using a Wavelet PCA Representation of Faces	A novel method was introduced which reduced error rate processing time.
	(Zifeng Li, Xiaao Tang)	Bayesian FR using SVM and Face Clustering	A hybrid algorithm combining Bayesian analysis and SVM was proposed which was found to be better than typical subspace methods.
2005	(Zhang, Guan, Wang, Liang, Quan)	FR in color image using PCA and FSVM	Color Segmentation was used to detect faces on frontal face image and better performance was measured.
	(Jun Qin, Zhong-shi He)	A SVM FR method based on Gabor Featured key Points	The new method comprising Gabor-Featured Key Points and SVM was used which highly improved the performance.
2006	(Praveen, Bhawani)	FR using Multiple Classifiers	A comparison of LDA, K Nearest Neighbor (KNN) and SVM was performed. SVM provided better results in real time with high accuracy.
2007	(Liu, Chen)	FR using Total Margin based Adaptive FSVM (TAF-SVM)	TAF-SVM was found to give high face recognition accuracy and smaller error variances.
2008	(Lihong, Ying, Yushi, Cheng, Yi)	FR using Multi Scale PCA (MS-PCA) and SVM	SVM was used with MS-PCA for dimension reduction. It provided good performance through Gabor features.
2009	(Lang, Xia, Wang)	FR using proximal SVM (PSVM)	To lower training time, PSVM was introduced which uses PCA.
	(Jiang, Zhou, Hou, Lin)	FR with Kernel Subclass Convex Hull (KSCH) sampling method	Execution speed was increased with high accuracy.
	(Andrew Y.ng, Honglak lee, Rajesh Ranganath, Roger)	CDBN for Scalable unsupervised Learning of Hierarchical representations.	The author proposed Convolutional DBN (CDBN) model which is translation invariant and supports efficient probabilistic inference.
2010	(liu, Tian, Peng, Li)	FR based on Least Square SVM (LS-SVM) and Particle Swarm Optimization (PSO)	This approach reduced convergence time and increased searching capability of algorithm. The improvement was found to be suiTable for facial expression recognition.
	(Wang, Liang)	FR using Fuzzy Rough Set and SVM	This algorithm bypass loss of information and provides superior recognition rate.

Table 5. Summary of Face Recognition Literature from Year 2010 till 2015

Year	Authors	Title	Explanation
2011	(Wang, Lan, Zhang, Gu)	FR based on PCA and SVM	PCA was used as feature extraction tool and SVM as classifier but found to be beneficial for small training data.
	(Zhong, Liu)	Bilinear Deep Learning for Image Classification	Bilinear DBN (BDBN) for semi-supervised image classification is proposed which provides better results than many Deep Architectures[16][20].
2012	(Honglak lee, Gary B. Huang, Erik Learned Miller]	Learning Hierarchical Representations for Face Verification with CDBN	CDBN is used to extract complementary representations and combine it with hand-crafted feature descriptors to achieve state-of-the art results.
2014	(Hoqiang Fan, Zhimin Cao, Yuning Jiang, Qi Yin)	Learning Deep Face Representation	The author proposes a new easy-to-implement deep learning framework and name it as Pyramid CNN which makes training procedure very fast and achieves high accuracy results.
	(Junlin Hu, Jiwen Lu, Yap-Peng Tan)	DDML for face Verification in the wild	Discriminative Deep Metric Learning (DDML) which beats many state of the arts in the field.
2015	(Omid Sakhi)	Face Detection in Matlab with Gabor Features and NN	Author classify given facial image as face or non-face using Gabor Features and NN using dataset [20].

3.1. Restricted Boltzmann Machine (RBM)

Since RBM possess hidden units so its probability distribution becomes function of two variables namely: input $\boldsymbol{x}$ and non-observed variables or hidden units h. Note that known variables or visible units $\boldsymbol{v}$ are the same as given input example $\boldsymbol{x}$.

$$P(x) = \sum_h P(x,h) = \sum_h \frac{e^{-Energy(x,h)}}{Z}$$

The free energy of the input (i.e. its un-normalized log-probability) is given as:

$$Free\ Energy = -log \sum_h e^{-Energy(x,h)}$$

This scalar energy is a measure of compatibility between observed variables X and unobserved variables h. Learning in EBM strives to modify the energy function such that it associates low energy to desirable configurations (correct values) and higher energies to undesirable configurations (incorrect values) (Deep Learning, 2016). In an RBM, the connections between visible layer and hidden or latent layer is such that the only interaction terms are between hidden units and visible units, but not between units of the same layer (see Figure 2). which illustrates the graphical structure of an RBM.

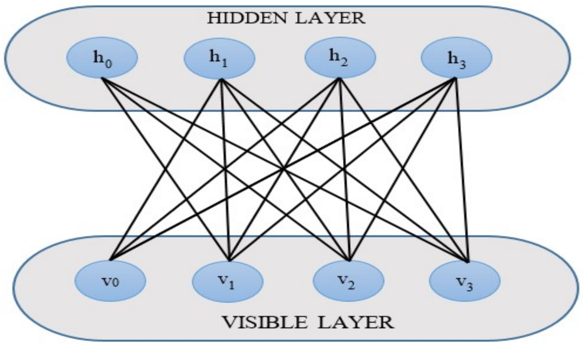


Figure 2. Structure of an RBM with 4 visible and 4 hidden units

The visible units of the RBM represent input features and hidden units can be viewed as feature detectors. Hence, it forms a bipartite graph with a set of visible units $v \in \{0,1\}^{nv}$ and a set of hidden units $h \in \{0,1\}^{nh}$ where nv and nh are the number of visible units and the number of hidden units respectively. As there is no input to input and hidden to hidden connections, its energy function is bilinear and is given as:

$$Energy(x,h) = -(b'x + c'h + h'Wx)$$

Where b' and c' are the biases vectors for visible units and hidden units respectively. Here apostrophe denotes transpose of a vector. W is the weight matrix of dimensions $[\mathrm{nv}, \mathrm{nh}]$ which represents symmetric weights between visible visible unit i and hidden unit j. Hence, the free energy of the input (i.e. its un-normalized log-probability) is given as (Youshua,2009) (LeCun et.al, 2006):

$$Free\ Energy = -(b'x + \sum_i log \sum_{h_i} e^{h_i W_i x})$$

Where W_i is the row vector which gives weights associated with i^{th} visible unit of the weight matrix W. And the normalizing constant Z is given as (LeCun et.al, 2006):

$$Z = \sum_x \sum_h e^{-Energy(x,h)}$$

Due to the bipartite property of RBM, input and latent variables are conditionally independent. The conditional probability $P\ (h|x)$ i.e. probability of the configurations of h for each given x is given as (Gan et.al, 2015):

$$P(h|x) = \prod_i P(h_i|x)$$

Similarly,

$$P(x|h) = \prod_j P(x_j|h)$$

RBM can be used to create deeper models by stacking them like Deep Restricted Boltzmann Machines (DRBBM) or DBNs. In case where visible and hidden units can have only binary values (i.e. 0 or 1), its conditional probabilities changes to:

$$P(h_i = 1|x) = \frac{e^{c_i + W_i x}}{1 + e^{c_i + W_i x}} = sigm(c_i + W_i x)$$

Where W_i is the row vector which gives weights associated with i^{th} visible unit of the weight matrix W. As input units and hidden units are playing symmetric role in the RBM model, hence from:

$$P(x_j = 1|h) = \frac{e^{b_j + W'_{.j} h}}{1 + e^{b_j + W'_{.j} h}} = sigm(b_j + W'_{.j} h)$$

3.2. Sigmoid Belief Network (SBN)

Sigmoid Belief Net is a Directed Acyclic Graph (DAG) which connects binary stochastic visible neurons with binary stochastic hidden layers. It is similar to RBM as it also forms a bipartite graph except that it has directed connections from the top layer to the input layer. Figure. 3 shows it graphical structure. SBN is a type of Bayesian Network and its conditional distribution is given as [9]:

$$P(x_i = 1|h^{(1)}) = sigm\left(b^{(0)} + W^{(1)^T} h^{(1)}\right)$$

Where x_i is the i^{th} input binary variable, $h^{(1)}$ is the 1st hidden layer, $b^{(0)}$ is bias vector for input layer and $W^{(1)}$ is weight matrix between visible units and the 1st hidden layer. The energy function of SBN is not linear as RBM. SBN gives directed explanation for samples generated in the visible layer, hence it specifies generative process to obtain data[9].

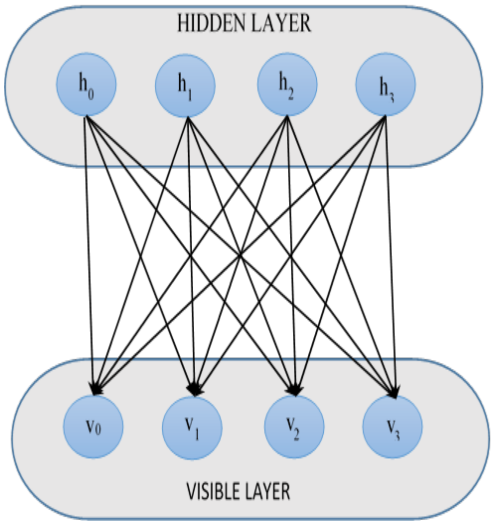


Figure 3. Structure of an SBN with 4 visible and 4 hidden units

3.3. DBN as a Stack of Generative RBM

Deep Belief Networks (DBN) is a brainchild of Geoffrey E. Hinton, who is a great computer scientist and is noTable for his work in Machine Learning (ML) and Artificial Neural Networks (ANN). Finding limitations of the type of models that can be represented efficiently by a single layer of hidden variables, Hinton proposed l layers deep model which represents the joint distribution between input vector (observed variables) x and l hidden layers h^kas follows [1]:

$$P\left(x, h^1, \ldots\ldots, h^l\right) = \left(\prod_{k=1}^{l-2} P\left(h^k \middle| h^{k+1}\right)\right) P\left(h^{l-1}, h^l\right)$$

Where x, the input vector which is considered as the 0^{th} hidden layer ($x = h^0$). $P\left(h^k \middle| h^{k+1}\right)$ is a conditional distribution associated with layer k for visible hidden units in an RBM (i.e. hidden units at level k are considered as visible units for level $k+1$). The top level RBM has the visible hidden joint distribution as $P(h^{l-1}, h^l)$. DBN is a generative model which is composed of RBM in the top layer l (which models joint distribution of undirected symmetric connections between visible-hidden units) and SBN in the rest of the layers $\{1, \ldots l-1\}$ (which models directed conditional distribution between visible-hidden units) .

Figure. 4 shows the structure of a DBN with 3 hidden layers $\{h^1, h^2, h^3\}$ and input features as $x = h^0$. DBNs are graphical probabilistic models trained in a greedy manner to extract high-level representation and abstractions in the data. Each layer in a DBN captures high-order correlations between the interactions of the units in the layer below.

DBNs are mostly trained to model the input distribution and generate new samples from it that's why they are categorized as Generative RBM. Generating new samples from an RBM is purely an unsupervised training. A generative model of RBM can either be trained in supervised or unsupervised way. Given a training set (x_i, y_i) where x_i denotes the i^{th} training example, $x_i \in \{1, \ldots., k\}$ and $y_i$is its corresponding target label $y_i \in \{1, \ldots., L\}$, a minimization of negative log-likelihood is used as:

$$R_{GEN}\left(x_i, y_i\right) = -\sum_{i=1}^{k} \log p(y_i, x_i)$$

In contrast to generative RBM there also exist discriminative model of RBM known as DRBM is used for classification where the ultimate goal is to get the correct label rather than model the distribution $p(x)$. Hence, the focus is to optimize $p(y|x)$ instead of $p(y, x)$:

$$R_{DISC}\left(x_i, y_i\right) = -\sum_{i=1}^{k} \log p(y_i | x_i)$$

DBN can be viewed as a stack of RBMs in the process of pre-training, the top layer forms an RBM. For example, consider a DBN with an input vector $(x = h^0)$ and hidden layers $\{h^1, h^2, h^3\}$. In the training of first layer $\{h^0, h^1\}$, DBN is considered to have only one layer, hence it is same as an RBM. In the training of second layer $\{h^1, h^2\}$, DBN is composed of two layers where the second layer is the top layer and forms an RBM, whereas the first layer behaves like a SBN. In the training of last layer $\{h^2, h^3\}$, the third layer forms an RBM and rest are SBNs. See Figure. 5 which illustrates the stacking procedure.

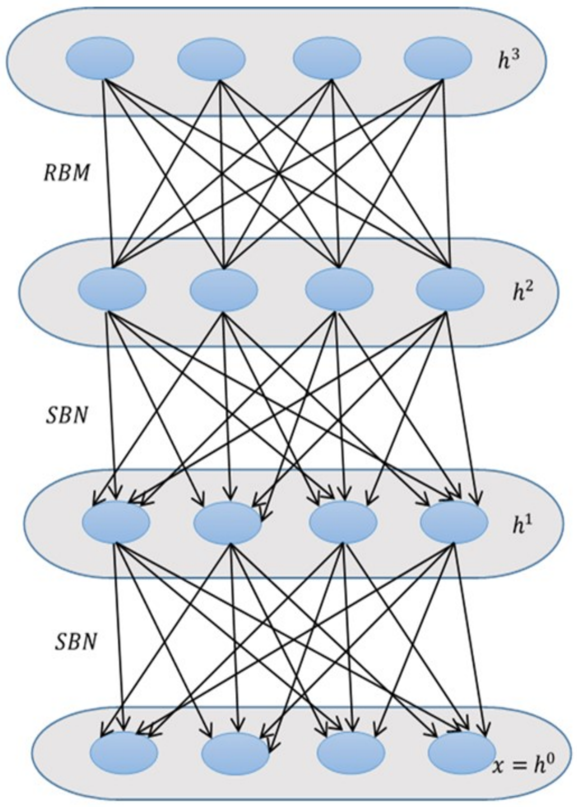


Figure 4. A DBN with three hidden layers, top layer forms an RBM and others form SBN

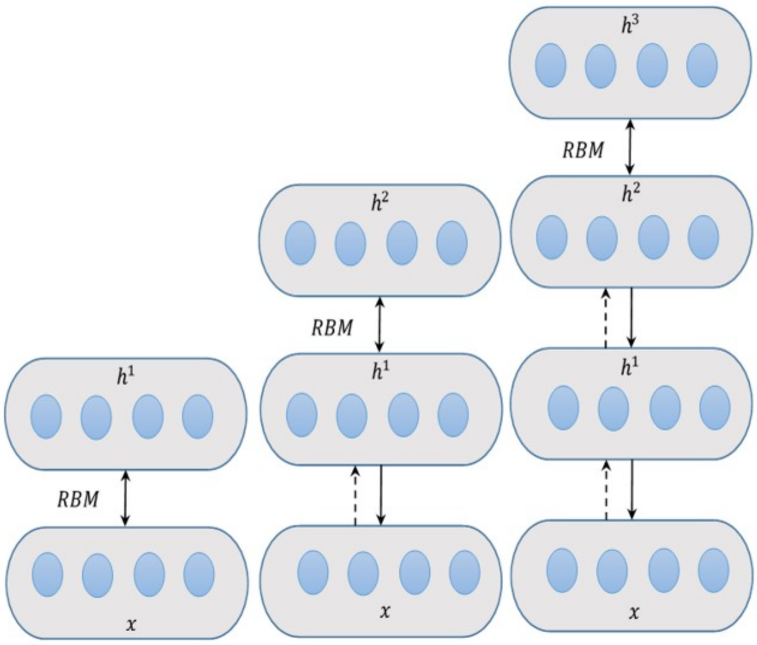


Figure 5. DBN as a stack of RBMs with layer-wise structure. The top layer always forms an RBM while the rest of the layers are SBNs. The generative path is shown with bold arcs and the recognition path is shown with dashed arcs.

4. Related work

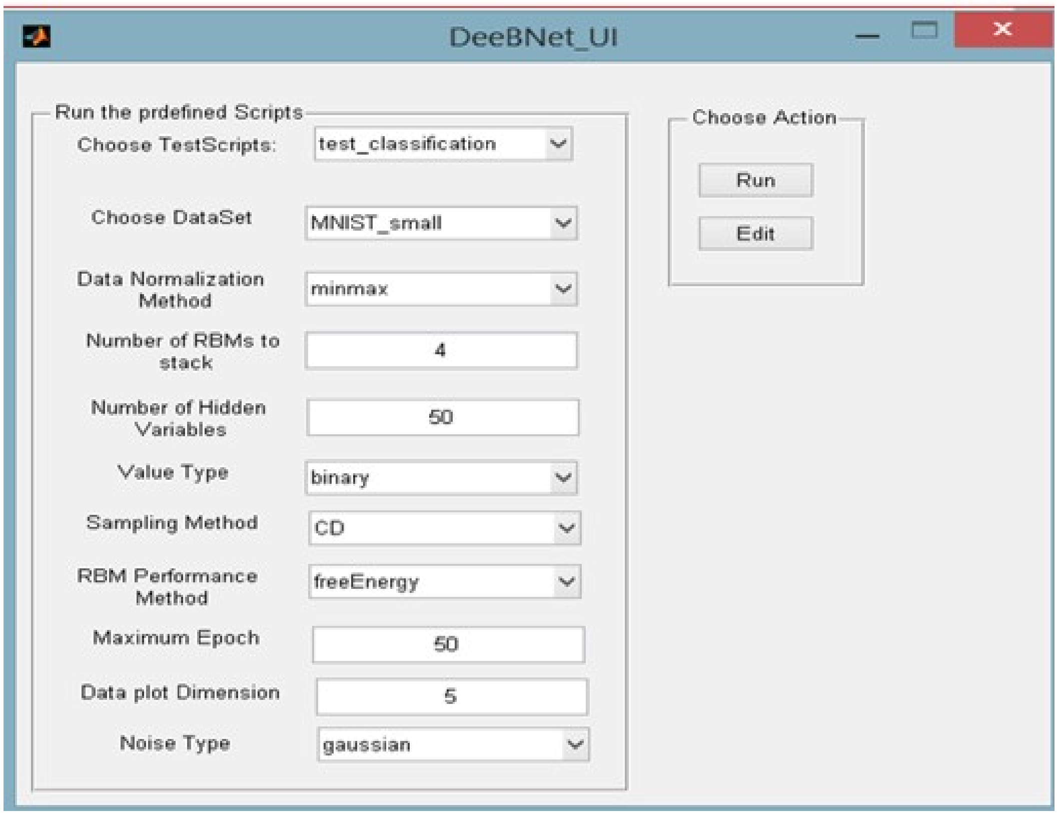
Our main work focuses on "Discriminative Deep Metric Learning for Face Verification in the Wild" (Hu, J., Lu,J. & Tan, Y.P.,2014)**.** The paper follows the DDML approach for classification and reports a competitive performance on the LFW and YTF datasets. The work identifies that facial images contains high-level information and thus non-linear transformation should be applied to capture its multiplex features. Moreover, when face images are captured with great variation in scenes, pose and expressions, the need to map face images in to a high-dimensional feature space is evident. The conventional *Mahalanobis distance* captures only the linear transformation ignoring the non-linear and high-level representations of the face images. Hence, the author proposed a *Discriminative Deep Metric Learning (DDML)* approach based on Deep Architecture which learns hierarchical non-linear distance metric to tackle these problems and uses back propagation algorithm to train the model (Lu. J., Hu.J & Tan, Y.P., 2017). The goal of Metric Learning approach is to learn a distance metric such that the distance between positive pair is minimized and that of negative pair is maximized. Many metric learning algorithms exist to tackle the problem of face verification.

For the DDML method, the Deep Neural Network (DNN) is used to capture the high-level representations of face pair by passing them to 3 layers of non-linear transformation. The DNN is trained using the Training set: $X = \{(x_i, x_j, \ell_{ij})\}$ where x_i and x_j are features extracted from a face pair and ℓ_{ij} is the corresponding pairwise label. The binary pairwise label ℓ_{ij} denotes similarity or dissimilarity between a given face pair x_i and x_j. To test the DDML, the author first extracts k feature descriptors for each face image and calculates k similarity scores to form a k–dimensional vector. In the end, the vector is averaged to get the final similarity for verification. The author provides the comparison of Discriminative Deep Metric Learning (DDML) vs. Discriminative Shadow Metric Learning (DSML model comprised of only one layer). The author uses different feature descriptors like *Dense Scale Invariant Feature Transform (DSIFT), Sparse Scale Invariant Feature Transform (SSIFT) and Local Binary Patterns (LBP)*. Results shows that Deep Metric Learning outperforms Shadow Metric Learning consistently with different feature descriptors. The reason for this is because DSML captures only the linear transformation whereas DDML learns non-linear relationship of samples. The author also compares DDML with state–of-the-art methods like Pairwise Constrained Component Analysis (PCCA), Fisher Vector Faces, and Cosine Similarity Metric Learning + Support Vector Machine (CSML+SVM), Pose Adaptive Filter (PAF). The paper shows the results with mean verification rate and standard error. The DDML approach with combined feature descriptors surpasses all existing algorithms. Similarly, in the end, the author compares DDML with different deep learning methods like Convolutional Deep Belief Network (CDBN) and Sparse Scale Invariant Feature Transform (SSIFT) with LFW dataset. The results reported with mean verification rate and standard error shows that the DDML approach with combined feature descriptors beats all other deep learning methods. The combined feature descriptors with DDML gives 90.68 ± 1.41 accuracy.

5. DeeBNet Toolbox

The DeeBNet toolbox is open source and freely available on the internet. The authors also presented a paper along with their toolbox namely, "**A brief survey on deep belief networks and introducing a new object oriented MATLAB toolbox (DeeBNet)**"[6]**.** The authors experimented with two datasets namely MNIST (Handwritten image dataset) and ISOLET (Speech dataset) and found complimentary results. This toolbox is an object-oriented implementation of Deep Belief Network features in Matlab. It has a complete documentation with test scripts available. DeeBNet toolbox can be used to: (1) Extract features from data without using label information (2) Reconstruct Data (3) Reduce Noise from Data (4) Classify data using Discriminative RBM (5) Generating data from Trained model. To get a better understanding of the *DeeBNet toolbox*, we developed a GUI which is like a wrapper on predefined scripts. It can be used to experiment with test scripts as shown in Figure. 6. It has options to change sampling methods, alter data normalization method and much more.

[6] Keyvanrad M. and Homayounpour M.(2014). DeeBNet toolbox. http://arxiv.org/


DeeBNet_UI

Run the prdefined Scripts

Choose TestScripts: test_classification

Choose DataSet: MNIST_small

Data Normalization Method: minmax

Number of RBMs to stack: 4

Number of Hidden Variables: 50

Value Type: binary

Sampling Method: CD

RBM Performance Method: freeEnergy

Maximum Epoch: 50

Data plot Dimension: 5

Noise Type: gaussian

Figure 6. GUI Developed for working with DeeBNet Toolbox

5.1. Sampling Methods

The DeeBNet toolbox provides an option to choose from different sampling methods to meet the requirements of our model. For this, it uses *SamplingMethodType* enumeration class which offers the following options:
GIBBS: It is a base class for all other sampling methods. It generates samples from an RBM model with random initialization samples for visible units. It is a common method for probabilistic inference and is well suited to deal with incomplete or missing information.
CD: *Contrastive Divergence (CD)* inherits features of Gibbs sampling but uses training data as initial value for visible units.
PCD: *Persistent Contrastive Divergence (PCD)* derives from Gibbs class and uses random samples for initialization in the first iteration. But contrary to normal CD, in the following iterations it does not start from scratch. Instead, it uses last chain state in the last update step.
FEPCD: *Free Energy in Persistent Contrastive Divergence (FEPCD)* derives from PCD and specifies criterion for selecting the best chain by using free energy of visible samples. Hence, gradient computation and generated samples are more precise as compared to PCD.

5.2. Visible Node Type

ValueType is an enumeration class in DeeBNet Toolbox which defines different forms of units in DBN. These types are: (1) BINARY: With 0 or 1 value. (2) PROBABILITY: With values in interval [0, 1]. (3) GAUSSIAN: Any real values with unit variance and zero mean.

5.3. Fine Tuning

It is a common strategy used in Deep Learning to make small adjustments in the trained network to improve its efficiency and get desired performance. Back Propagation and other discriminative algorithms are applied for Fine-tuning the pre-trained DBN weights and parameters. This is helpful when training data is limited.

5.4. Class Organization

The DeeBNet toolbox has two packages and some classes to deal with data and to define different RBMs and DBN. It has a class called *DataStore* which encapsulates variables regarding data (e.g. dataMean, dataStd, dataMin, etc.). It also has containers for storing training data, testing data and validation data along with their corresponding labels. Furthermore, it has functions for normalizing, shuffling and plotting data. It has three classes regarding RBM. The abstract *RBM* class defines all necessary functionalities for the inherited RBM classes like training method and features. The inherited classes from RBM are *GenerativeRBM* and *DiscriminativeRBM*. The *GenerativeRBM* class has the features of generative model and contains methods like *train*, *getFeature, reconstructData, generateData*, etc. It uses data without their labels. The *DiscriminativeRBM* has the capability to discriminate and classify data. For this, it has two different methods which are *generateClass* and *predictClass*. The *generateClass* generates data with a specified label whereas *predictClass* predicts class number or label of input data. It requires data with their labels for training. Another supporting class is *RBMParameters* which contains all parameters of an RBM like learning rate, epochs, weight matrix, biases, etc. It also has an interface class called *Sampling* class which binds all sampling methods. The *Gibbs* class implements it and is a parent class for all other learning methods. The classes which inherits from *Gibbs* are *Cd* and *Pcd. FEPcd* is another class which inherits from *Pcd* class. These classes define sampling methods that are used in RBM. It also has an important class called *DBN* which creates an arbitrary DBN based on all other classes (i.e. *RBM, DataStore, RBMParameters*, etc.). A DBN can be used as an **autoEncoder** (using *GenerativeRBM*) or as a **Classifier** (using *DiscriminativeRBM*). It has some efficient methods like *train, getFeature, addRBM, getOutput, backpropagation*, etc.

6. Methodology

To classify facial image to their best matching predefined category or label using Deep Belief Networks, we use LFW dataset whose sample is shown in Figure. 7. Following methods has been adopted for classification and Similarity Analysis with matched face (Hu et.al, 2014) (Lu, 2017).

Figure 7. Sample Images from LFW Dataset

6.1. Cross Validation

To experiment with LFW dataset for $N - ways$ face classification or face identificación, input was provided in two ways: (1) SINGLE IMAGE*:* To perceive the performance of N –way face classifier visually, a single image was used as test image. In this mode, a test image and its corresponding matched images are shown before and after back propagation. (2) BATCH PROCESSING*:* In batch processing mode, a series of test images are given to the pre-trained DBN to classify and label them. Later the actual and classified labels are compared to see the performance of classifier and for statistical analysis. The information acquired through batch processing will be used for plotting *Receiver Operating Characteristic (ROC)* curve and *Precision-Recall curve*.

For batch processing, a dataset has to be divided into training and testing sets. Cross validation is a set of model evaluation techniques in which the dataset is divided into two disjoint sets namely, training and testing. The model is trained using the training subset and for evaluating its performance, it is tested on new data or unseen examples i.e. testing set. The three popular methods for cross validation are: (1) Holdout Method (2) K- Fold Method (3) Leave One out Method. Hold out method is the simplest method in which the dataset is randomly divided into disjoint training and testing sets only once. In K-fold method, the dataset is divided into K-subsets. Hence, the model is evaluated K times using one subset for testing and rest of the subsets for training. The third Leave One out method is a special case of K-fold cross validation in which the size of subset is just one item. Hence, the number of subsets is $K = N$, where N is the number of data items in the dataset. We have used the K *–fold method* for cross-validation to evaluate the performance of N-way Classifier. The LFW dataset is divided into 12 subsets depending on the number of images per individual. For example, the individual having only one image constitute a set in which training and testing data is same. Likewise, sets are formed for 2,3,4,5,6,7-10,11-20, 21-26,48,53 and 83 images per individual.

6.2. Data Matrix and Label Vectors

To pre-train the DBN, input is to be provided in vector form. Since the LFW dataset contains $2D$ intensity images of 250×250 pixels. The image is fetched and resized to 60×60 pixels. It is converted to row vector and saved into Data Matrix as shown in Figure 8. Hence, the dataset is converted in to a matrix containing images as row vector. A column vector is also used accordingly to assign the corresponding labels as shown in Figure. 9. Given K images of N different individuals where $K > N$ as there are one or more images per individual. Hence, there are N unique labels corresponding to N individual's images which makes it an N*-ways classification problem*. Instead of using $2D$ intensity images as raw input, extracted features must be used to get feature descriptors for facial images to train the DBN for better performance. We also experiment with *LFW Deep Funneled* version of Data set which provides aligned facial images for LFW dataset. We have used a small subset of *LFW deep funneled* version because of the high computational time taken for the training of a large dataset. The *small LFW deep funneled* dataset contains only people whose name start with the letter *A*. The image fetched from *LFW Deep Funneled* folder is changed to grayscale, cropped from center which reduces its size to $[150 \times 150]$. It is then resized to $[60 \times 60]$ pixels which forms a 3600 *row vector* for each image. PCA is applied for dimension reduction which reduces each row vector of image in *Data Matrix* to 500 dimensions as depicted in Figure 10.

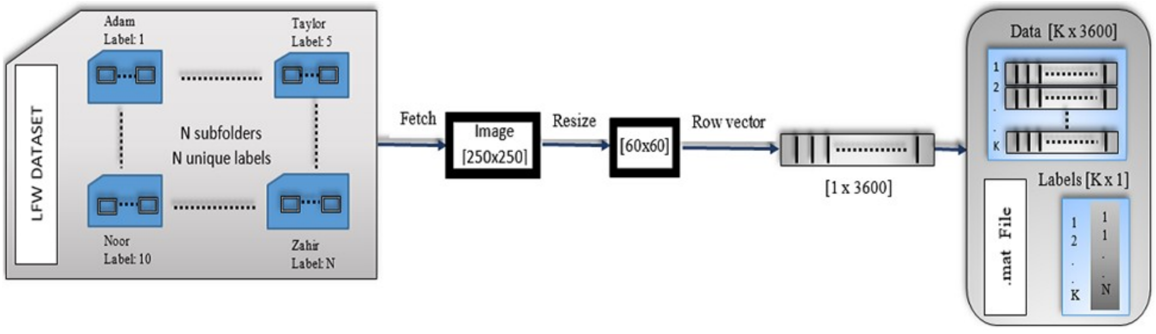


Figure 8. A pictorial representation of the conversion of LFW Dataset into Data Matrix with multiple labels

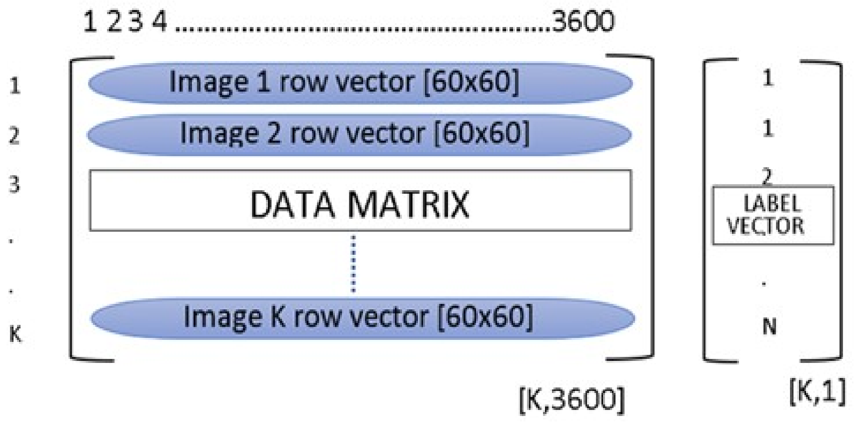

1 2 3 4 3600
1
2
3
K
Image 1 row vector [60x60]
Image 2 row vector [60x60]
DATA MATRIX
Image K row vector [60x60]
1
1
2
LABEL VECTOR
N
[K,3600]
[K,1]

Figure 9. Design of the Data Matrix and Label Vector

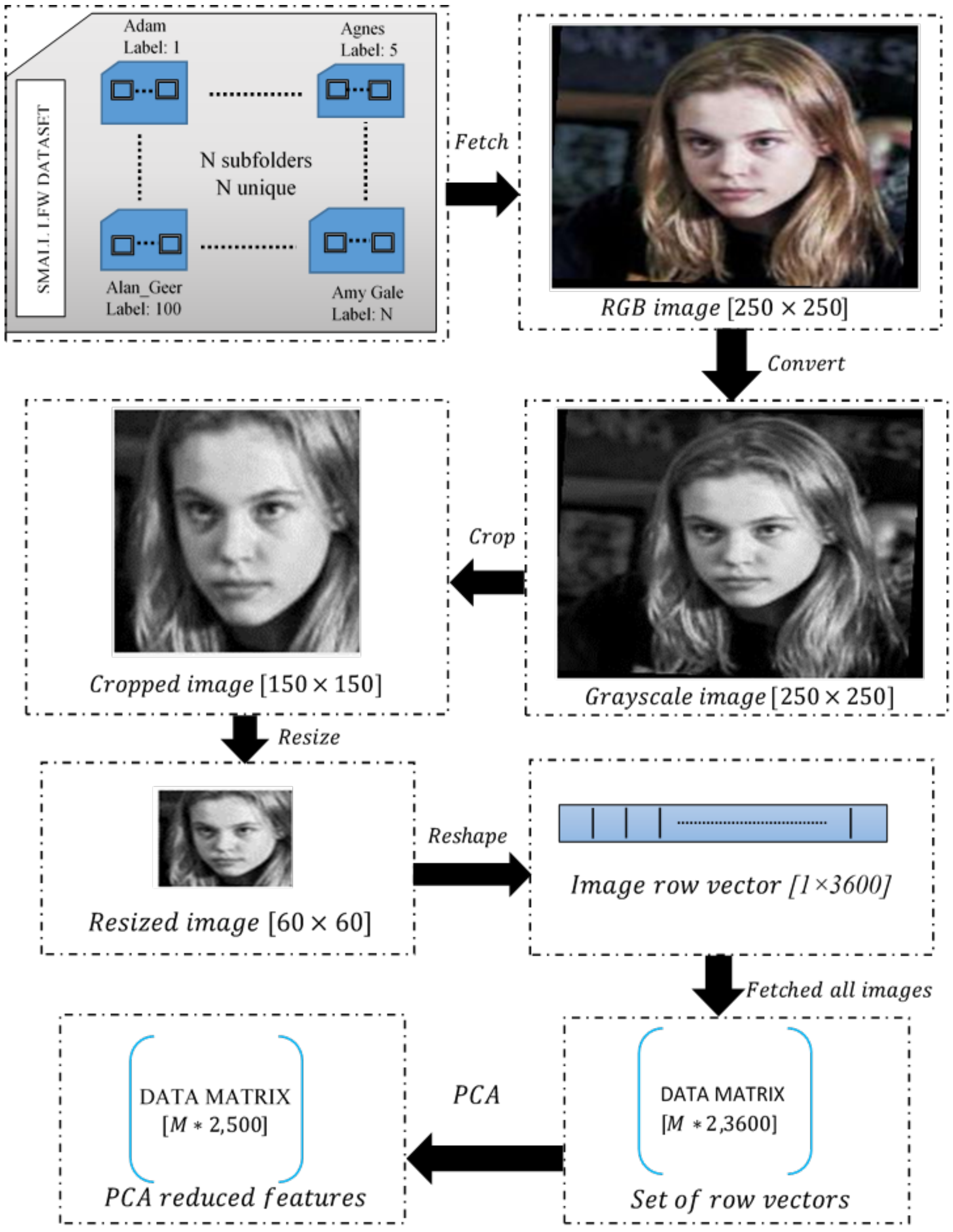


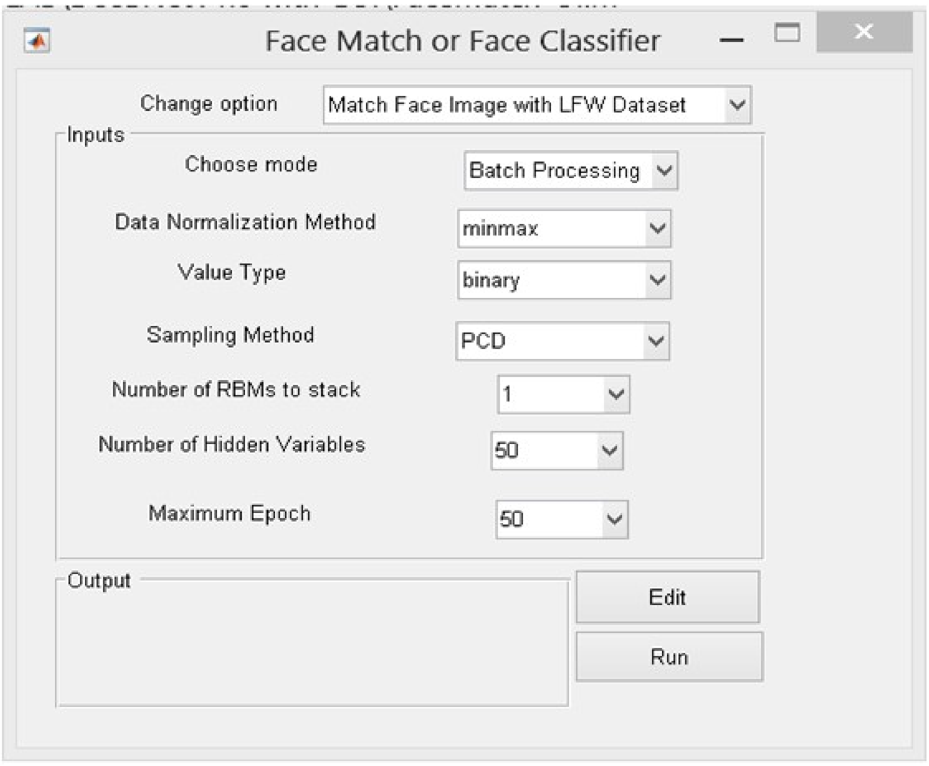
Figure 10. Flow of feature extraction with PCA

6.3. DBN Settings as Classifier

Recall that a DBN is a generative model which is composed of stacking of *Restricted Boltzmann Machines (RBMs*) in two ways:

- ✓ *Generative RBM*: Used for feature extraction and reconstruction of data.
- ✓ *Discriminative RBM*: Used to classify and seek out differences in data.

Hence, a classifier DBN is composed of layers of Generative RBM but uses a Discriminative RBM in last layer as a classifier RBM. The Classifier DBN aims to find labels from input data. As PCA has reduced the $2D$ intensity image of $[60 \times 60]$ into a 500 dimensional vector. Hence, the input layer of the DBN has 500 visible units. The DBN maps the 500 visible features to 250 hidden units in high-dimensional space. Hence, an *autoEncoder* DBN generates a 250 dimensional feature vector. *Contrastive Divergence* method is used for sampling in RBMs. A GUI is also created for Face Classifier as shown in Figure. 11. Options has been provided to change number of RBMs to stack, to experiment with different sampling methods and value type. Moreover, it allows to switch from Batch Processing to Single Image testing mode.


Face Match or Face Classifier

Change option: Match Face Image with LFW Dataset

Inputs

Choose mode: Batch Processing

Data Normalization Method: minmax

Value Type: binary

Sampling Method: PCD

Number of RBMs to stack: 1

Number of Hidden Variables: 50

Maximum Epoch: 50

Output

Figure 11 Front-End of Face Classifier

6.4. Training and Testing for Similarity Analysis

A classifier DBN is pre-trained with training Data Matrix. To fine-tune the network, labels of the training data are used to back propagate the error through the network and adjust weights and parameters. Hence, when an input or test image is presented to a pre-trained DBN, it compares it with the training data and predicts label or class accordingly as shown in Figure. 12. To Determine whether the given image pair is similar or dissimilar we turned to *Image Restricted* and *Image Unrestricted* setting on LFW dataset. *Image Restricted* setting uses only pairwise label information ($\ell_{ij} = 1$ if the pair matches, else $\ell_{ij} = 0$ where x_i and x_j are images of a given pair $P(x_i, x_j)$). On the other hand, *Image Unrestricted* setting on LFW dataset can use extra information (like information that to whom the given pair of images belong) along with pairwise label ℓ_{ij}. This way the n-way

Face classification turns to a binary classification problem. This way each facial image pair $P(x_i, x_j)$ is treated as a single training example and images of a given pair are placed in consecutive manner in the *Data Matrix* as demonstrated in Figure. 13. But the problem raised here that what labels are given to DBN for training each image in a pair $P(x_i\ x_j)$ which are placed consecutively. The solution to this problem lies in the two forms: (1) Supervised Training (2) Unsupervised Training.

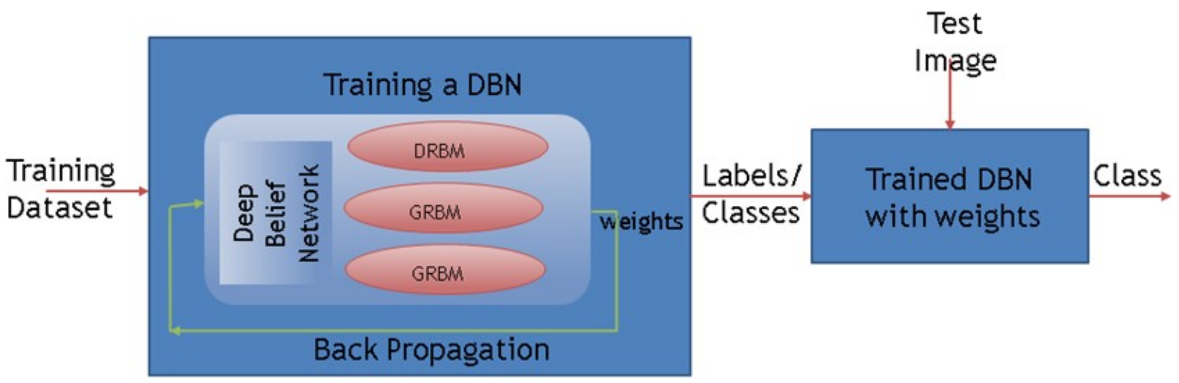


Figure 12. Workflow of DBN Classifier

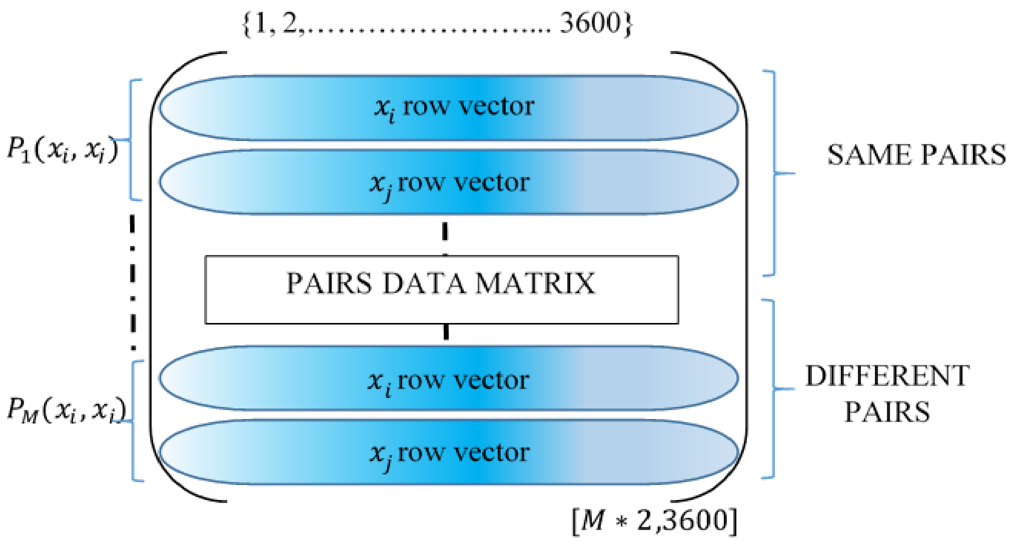


Figure 13. Training DBN using pair images

6.4.1. Design of N-Way Classifier based on Supervised Training

Train the DBN as an N-way face classifier by providing normal labels {1,2,3…N.} under the supervised training. For testing, give images in the form of pair separately to the pre-trained DBN to classify and label it. In the end, the labels for a given pair are matched. If they are same, assign label $\ell_{ij} = 1$ to pair $P(x_i, x_j)$, otherwise assign $\ell_{ij} = 0$. The flow of supervised approach is showcased in Figure. 14. This approach uses conversion of N-labels to binary labels. But this approach does not follow *Image Restricted settings* on LFW dataset as we have used N-labels information for each individual beside pairwise labels. The Output of this supervised training is demonstrated in GUI of Figure. 15. Which brings all matched results corresponding to an input image.

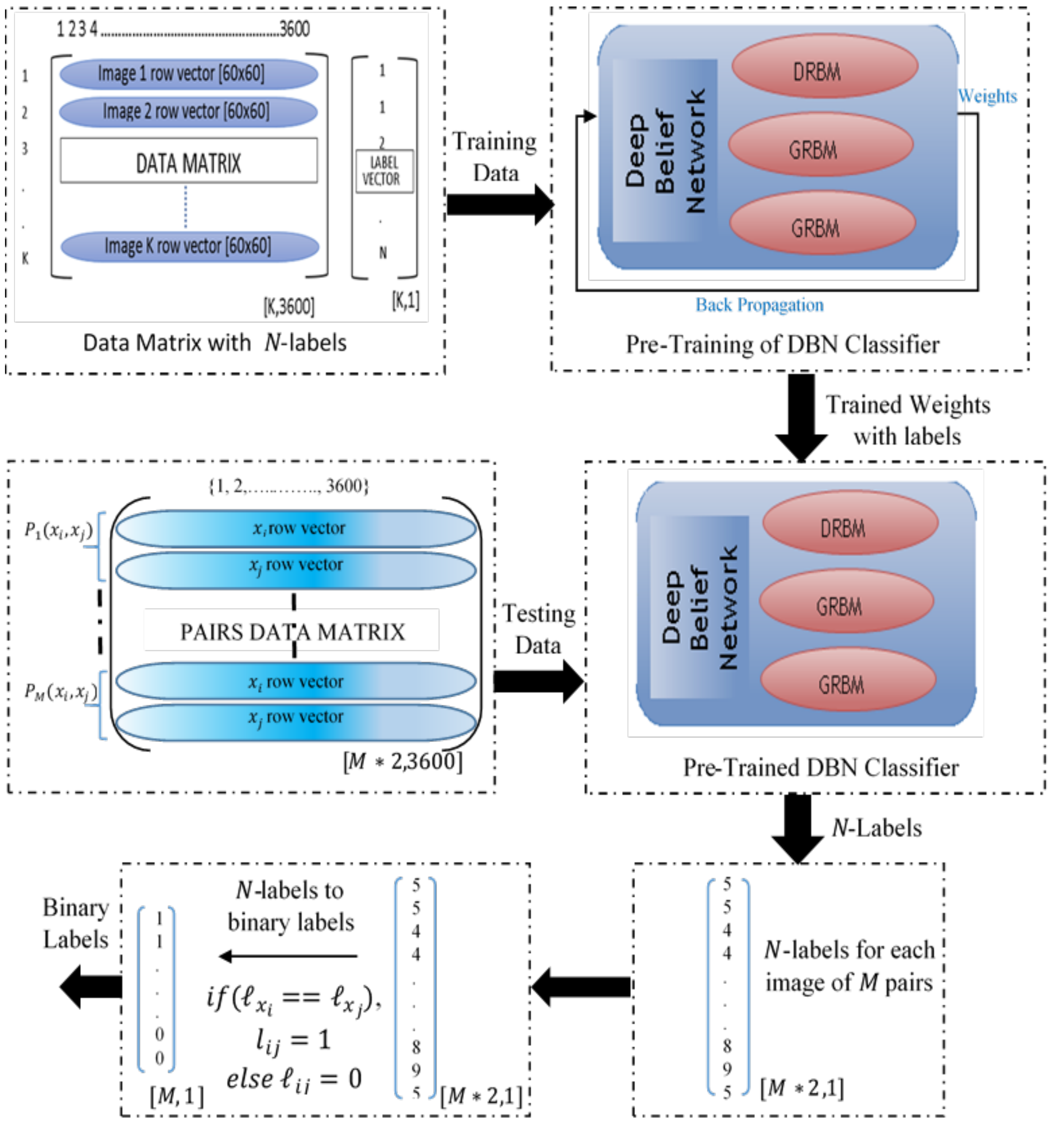

1 2 3 4 ...3600
Image 1 row vector [60x60]
Image 2 row vector [60x60]
DATA MATRIX
Image K row vector [60x60]
LABEL VECTOR
[K,3600]
[K,1]
Data Matrix with N-labels
Training Data
Deep Belief Network
DRBM
GRBM
GRBM
Weights
Back Propagation
Pre-Training of DBN Classifier
Trained Weights with labels
{1, 2,........, 3600}
$P_1(x_i,x_j)$
x_i row vector
x_j row vector
PAIRS DATA MATRIX
$P_M(x_i,x_j)$
[M * 2,3600]
Testing Data
Pre-Trained DBN Classifier
N-Labels
N-labels for each image of M pairs
[M * 2,1]
N-labels to binary labels
$if(\ell_{x_i} == \ell_{x_j})$, $l_{ij} = 1$ $else\ \ell_{ij} = 0$
[M,1]
Binary Labels

Figure 14. DBN Supervised Training and Transforming N-Way classification problem to 1-Way

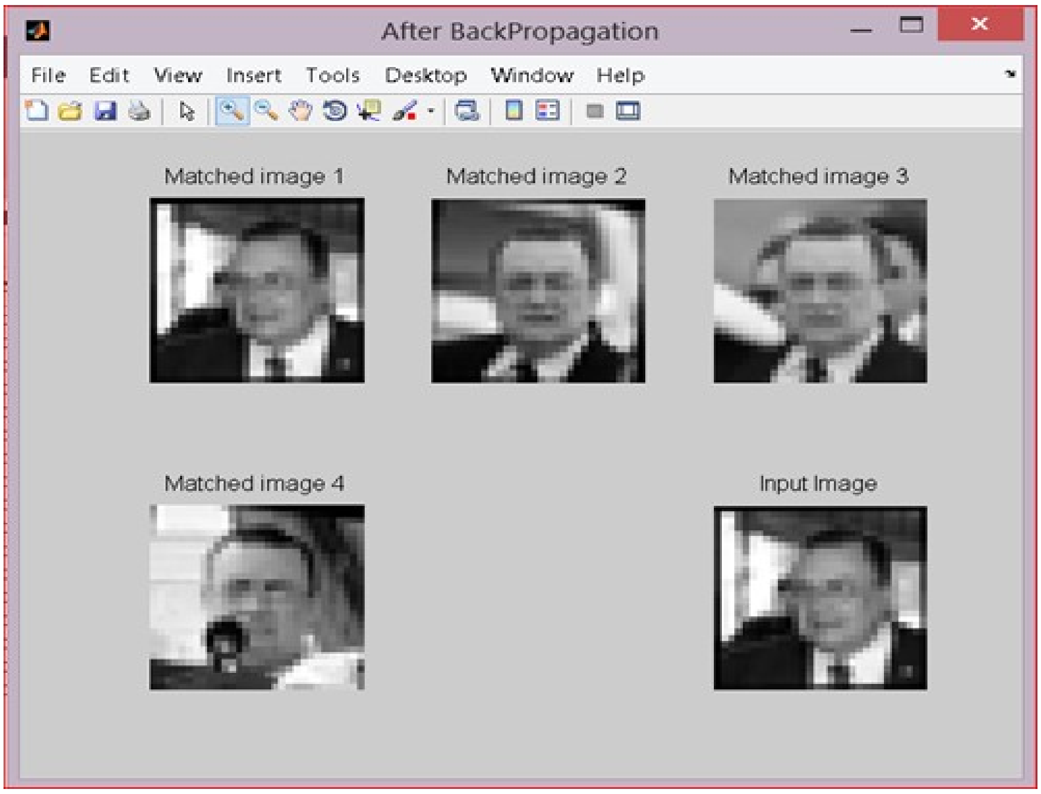


Figure 15. Sample Results from proposed N-way face classifier

6.4.2. Design of autoEncoder Based on Un-Supervised Training

For unsupervised training approach, a DBN can be used as an *autoEncoder* to encode or map each image into a high-dimensional space without labels. The high-level representation for each image is treated as a *feature vector* fv for that image. Lastly, the *distance metric* (*or similarity measure*) is applied on feature vectors $\left(fv_i, fv_j\right)$ of a given pair $P(x_i, x_j)$ to determine a matched or mismatched pair (Lu et.al, 2017) . The workflow of the unsupervised training approach is shown in Figure 16. The *Similarity Metric DBN autoEncoder* approach uses conversion of *distance metric* to binary labels (El-Syed & Aboelwafa 2012) (El-Syed & Hamed 2015) . This unsupervised training approach follows *Image Restricted settings* on LFW dataset as we have used only pairwise label for a given matched or mismatched pair.

7. Results Comparison and Discussion

We provide following comparison and analysis based on small LFW dataset:

1. Comparison of Supervised and Unsupervised DBN Classifier
2. Comparison of DBN Classifier and DBN Auto-encoder with and without PCA Features
3. Comparison of DBN Auto-encoder with Euclidean and Square Euclidean Distance

The comparison of N-way DBN face Classifier (Supervised training; Image unrestricted Setting) with Similarity Metric DBN autoEncoder (Unsupervised training; Image restricted setting) as a function of number of DBN layers is stated in the form of *mean verification rate with standard error* in Table 6, when PCA features are used. It has been observed that the *DBN autoEncoder* produces much better results as compared to the DBN classifier. This is because of the reason that DBN is basically a *generative* model which is used for feature extraction, reconstruction of data, removing noise from data, etc. A DBN can also be a good classifier, if it has to classify only a small number of classes. Moreover, the performance of DBN can be increased by refining its parameters like *Sampling Method, Number of hidden units, Number of Layers,* etc. Improvement in feature extraction process can also result in the betterment of these approaches. Clear enhancement in the proposed models has been seen by employing *PCA features* instead of raw 2D image's *intensity in both N-way DBN Face Classifier* and *Similarity Metric DBN autoEncoder*.

Table 6. Mean Classification Error with different number of layers

Number of Layers (NoL)	**1**	**2**	**3**
N-way DBN Face Classifier	58.48 ± 0.12	56.05 ± 0.135	56.21 ± 0.18
Similarity Metric DBN autoEncoder (EUC)	72.01 ± 0.005	59.03 ± 0.15	59.34 ± 0.182

Table 7. reports the results of this comparison. Using combination of other features will surely increase the verification rate of both the models. Finally, we experimented with distance measures (or similarity metric) namely Euclidean or squared Euclidean and recorded the results with mean verification rate in Table 8. This Table also reports computational time with different number of layers. The hardware configuration to compute time comprises a $2.30\ G-Hz$ CPU and *4 GB* RAM with *64-bit* Operating System (OS). For each fold, the DBN takes 60 seconds with one layer (does not include PCA feature extraction time). With each layer added, the training time (includes back propagation as well) increases. Here we have reported time for a small dataset. But as the training set increases, its training time also doubles. The testing time depends on the size of testing data.

Table 7 Mean classification Rate and Standard Error with and without PCA Features

Number of Layers (NoL)	**1**	**2**	**3**
DBN Face Classifier	58.48 ± 0.12	56.05 ± 0.135	56.21 ± 0.18
DBN Face Classifier (Without PCA)	38.97 ± 0.13	38.97 ± 0.13	38.97 ± 0.12
DBN autoEncoder	72.01 ± 0.005	59.03 ± 0.15	59.34 ± 0.182
DBN autoEncoder (Without PCA)	61.02 ± 0.13	61.02 ± 0.13	61.02 ± 0.13

Table 8 Mean Verification Rate and Standard Error of DBN autoEncoder

Number of Layers (NoL)	**1**	**2**	**3**
DBN autoEncoder (Sq-Euclidean)	72.44 ± 0.181	55.45 ± 0.104	51.30 ± 0.15
DBN autoEncoder (Euclidean)	72.01 ± 0.005	59.03 ± 0.15	59.34 ± 0.182
Computational Time (sec)	**60**	**90**	**120**

8. Conclusion and Future work

In this work, we learn and experiment with DBN and propose design of two new DBN models, namely *N-way DBN face Classifier* and *Similarity Metric DBN autoEncoder* for *facial similarity* (or Face Verification). The proposed approaches work with both *Image Unrestricted* and *Image Restricted* settings on the small LFW dataset. Our methods do not produce competing results as many state-of-the-art Face Verification algorithm due to small dataset usage and many improvements needed specially at feature extraction stage. With Advancement in Feature extraction, the proposed approaches might achieve competitive verification performance. For example, we can extract different features (SIFT, HOG, LBP, SURF, etc.) for a given image and combine them into a new feature vector which can be reduced by PCA. The combined reduced feature vector can then be used as an input to DBN, resulting in the enhanced performance of the proposed approaches. Moreover, the proposed models can be tested with different parameters (like *Sampling Method*, *Number of Hidden Units, Number of Layers,* etc.) to find out the improvements.

References

Bedre, J. S., & Shubhangi, S. (2012). Comparative Study of Face Recognition Techniques: *A Review, Emerging Trends in Computer Science and Information Technology*.

Deep Learning. (2016). Restricted Boltzmann Machine (RBM). http://deeplearning.net/.

El-Sayed, M.A., & Hamed, K. (2015). *Study of Similarity Measures with Linear Discriminant Analysis for Face Recognition. Journal of Software Engineering and Applications.*Vol. 8(9). pp. 478-488.

El Sayed, M.A., & Aboelwafa, N. (2012). *Study of Face Recognition Approach Based on Similarity Measures. International Journal of Computer Science Issues (IJCSi).* Vol. 9(5). pp. 133-139.

Gan, Z., Henao, R., Carlson, D., & Carin, L. (2015). *Learning deep sigmoid belief networks with data augmentation*, in AISTATS.

Hinton, D. (2009). Deep Belief Networks. http://www.scholarpedia.org

Hu, J., Lu, J. & Tan, Y. P. (2014). *Discriminative Deep Metric Learning for Face Verification in the Wild, In* Proc. IEEE Conference on Computer Vision and Pattern Recognition. CVPR. pp. 1875-1882.

Ian, G. F., Yoshua, B., Aaron, C. (2016). *Deep Learning*, Cambridge, Massachusetts, United States: The MIT Press.

LeCun, Y., Chopra, S., Hadsell, R., Ranzato, M., Huang, F. (2006). *A tutorial on energy-based learning,* Cambridge, Massachusetts, United States: The MIT Press.

Lu, J., Hu, J. & Tan, Y. P. (2017). *Discriminative Deep Metric Learning for Face and Kinship Verification, IEEE Transactions on Image Processing*. 26(9). pp 4269-4282.

Pandya, J. M., Rathod, D. & Jadav, J. J. (2013). *A Survey of Face Recognition approach, International Journal* of *Engineering Research* and *Applications* (*IJERA*). 3(1). pp. 632-635.

Youshua, B. (2009). *Learning Deep Architectures for AI. Foundations and Trends in Machine Learning, Vol. 2,* pp. 1-127.

Zhao, W., Chellappa, R., Rosenfeld, A., & Phillips P. J. (2003). *Face Recognition: A Literature Survey. ACM Computing Surveys.* 35(4). pp. 399-458.

Humera TARIQ received her B.E. in Electrical (1999), MCS in Computer Science (2003) and PhD in Computer Science (2015). Currently she is assistant professor of Computer Graphics, Vision and Image Processing in Department of Computer Science, Faculty of Sciences, University of Karachi, Pakistan. Her current research interests include different aspects of Artificial Intelligence applied on Outdoor Images, Human Images, Bio Medical Images and Documents. She is professional member of ACM and ACMSIGGRAPH since 2014.

Zainab PARVEEN received her BSCS. in Computer Science (2015) from Department of Computer Science, University of Karachi, Pakistan. Zainab Stood First Class First among 200 students of Batch 2012-2015 in BSCS. Her current research interest includes Deep Learning.