A Novel Integrated Face Verification Algorithm Selecting Feature-Rich Frames From a Video Sequence

[1]A.Lowkya , [2]Mr.D.Ravi Krishna Reddy
[1]M.Tech student (P.G scholar), [2]professor
[1]Deptartment of Electronics andCommunication,
Gudlavalleru Engineering College, Gudlvalleru, India-521356

Abstract: However, due to the widespread use of web-cams and mobile devices embedded with a camera, it is now possible to realize facial video recognition, rather than resorting to just still images. In fact, facial video recognition offers many advantages over still image recognition; these include the potential of boosting the system accuracy and deterring spoof attacks. Deep learning has recently achieved very promising results in a wide range of areas such as computer vision, speech recognition and natural language processing. It aims to learn hierarchical representations of data by using deep architecture models. In this paper, we propose a novel face verification algorithm, which starts with selecting feature-rich frames from a video sequence using Multi-wavelet transform and entropy computation. Frame selection is followed by representation learning-based feature extraction, where three contributions are presented: 1) deep learning architecture, which is a combination of stacked denoising sparse autoencoder (SDAE) and deep Boltzmann machine (DBM); 2) formulation for joint representation in an autoencoder; and 3) updating the loss function of DBM by including sparse and low rank regularization. Finally, a multilayer neural network is used as the classifier to obtain the verification decision. The results are tested on the YouTube Databases.

Index Terms— Deep learning, auto encoder, deep Boltzmann machine, face recognition, frame selection, Multi-Wavelet Transform.

I.INTRODUCTION

VIDEO face reputation has emerged as distinctly big in surveillance eventualities. For instance, more than 80,000 people were identified and proven at some stage in the 2008 Beijing Olympics with the help of face recognition in videos [1]. With improvements in technology, video capturing gadgets are reachable to a huge variety of humans within the phones and tablets. In unconstrained scenarios, videos captured by such devices may also be used by law enforcement agencies. Therefore, there is a high motivation to utilize video data to perform accurate face recognition. Fig. 1 shows frames from video clips in which the face regions have been detected and cropped. While a single frame from a video can only capture limited information, multiple frames capture a lot of information about the face pertaining to its appearance under the effect of common covariates such as pose, illumination, and expression. By utilizing the large variety of information present in a video, a robust and comprehensive representation of a face can be extracted and accuracy can be improved

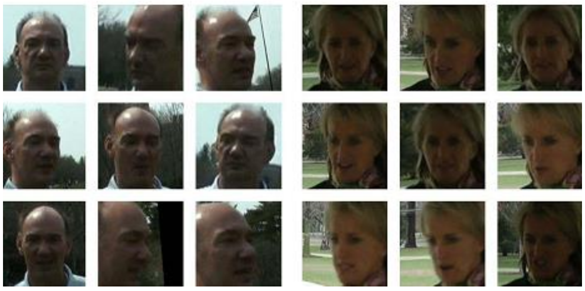


Fig.1: a subset of frames illustrating the quantity of information found in a video. A single video can seize a topic's face below specific pose, expression, and illumination versions. While a few frames can be enormously useful for face recognition, others may be unfavorable to overall performance. Images are frames from the PaSC database [2].

Video face recognition algorithms can broadly be categorized into two kinds:

(a) set-primarily based and

(b) Sequence based [2].

The set-primarily based approaches keep in mind a video as a fixed of images (frames) that are then modeled and coupled the use of a spread of methodologies. These tactics won't make use of the temporal statistics contained in the video, i.e. The order of frames inside the unique video may not count number. On the alternative hand, collection-based totally processes are in particular designed to utilize temporal information of the video. These methods model the video as a series of photos and apply series classification strategies for reputation.

For comparison, the outcomes are generally said on benchmark databases consisting of the Honda UCSD database [7], YouTube face database (YTF) [3], and these days developed Point and Shoot Challenge (PaSC) database [2]. As shown in Table I, existing algorithms have attained high performance on YouTube video face database [3]. However, the protocol of this database commonly requires reporting the consequences at equal blunders price (EER) [2]. From an implementation perspective, the algorithms are required to reduce fake be given rate (FAR) or false reject charge (FRR).

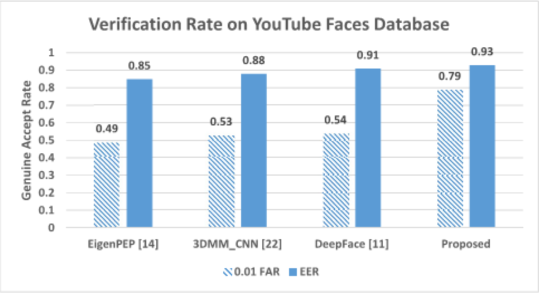


Fig2: Summarizing the performance of some of the best performing face verification algorithms on the YouTube faces database [3]. It is evident that there is a huge gap in the performance at low false accepts rates as compared to performance at EER. We showcase that the proposed algorithm performs well even at a low false accept rate.

Therefore, it is our assertion that there is a significant scope of improvement in the performance of video face recognition and additional research is required, especially focusing at lower false accept rates

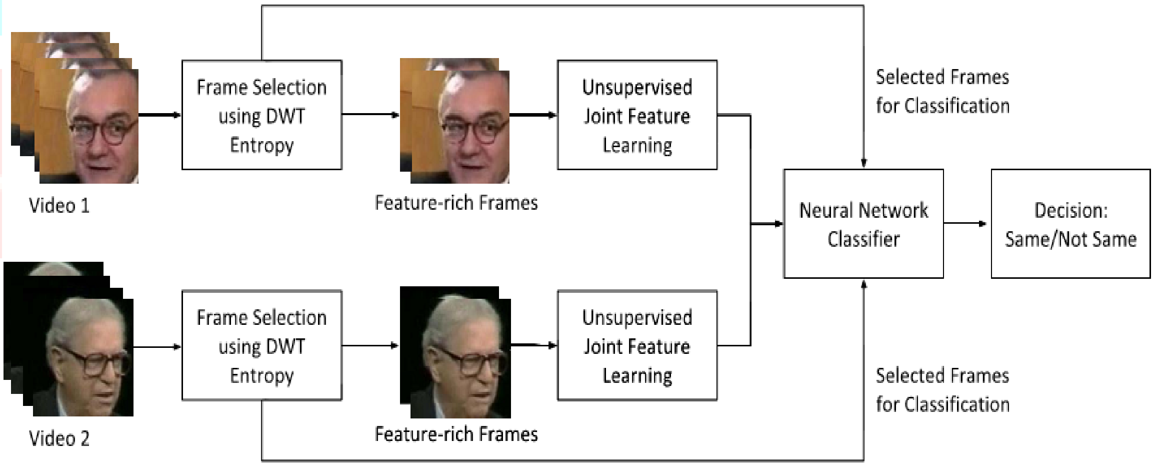

Video 1
Frame Selection using DWT Entropy
Feature-rich Frames
Unsupervised Joint Feature Learning
Selected Frames for Classification
Neural Network Classifier
Decision: Same/Not Same
Video 2
Frame Selection using DWT Entropy
Feature-rich Frames
Unsupervised Joint Feature Learning
Selected Frames for Classification

Fig.3: Proposed face recognition algorithm

II. LITERATURE SURVEY

[1] Safeguards Beijing: Facial recognition technology (FRT) has emerged as an attractive solution to address many contemporary needs for identification and the verification of identity claims. It brings together the promise of other biometric

Systems, which attempt to tie identity to individually distinctive features of the body, and the more familiar functionality of visual surveillance systems. This report develops a socio-political analysis that bridges the technical and social-scientific literatures on FRT and addresses the unique challenges and concerns that attend its development, evaluation, and specific operational uses, contexts, and goals.

[2] J. Beveridge et al Inexpensive “point-and-shoot” camera technology has combined with social network technology to give the general population a motivation to use face recognition technology. Users expect a lot; they want to snap pictures, shoot videos, upload, and have their friends, family and acquaintances more-or-less automatically recognized. Despite the apparent simplicity of the problem, face recognition in this context is hard. Roughly speaking, failure rates in the 4 to 8 out of 10 range are common. In contrast, error rates drop to roughly 1 in 1,000 for well controlled imagery.

The face images, videos, data, and associated metadata for the PaSC are available upon request. The support software including the Cohort LDA and LRPCA baseline algorithms and scoring code are downloadable through the web. We will generate and maintain a curated website where groups working on the PaSC may submit results.

[3] L. Wolf, T. Hassner, and I. Maoz, Recognizing faces in unconstrained videos is a task of mounting importance. While obviously related to face recognition in still images, it has its own unique characteristics and algorithmic requirements. Over the years several methods have been suggested for this problem, and a few benchmark data sets have been assembled to facilitate its study. However, there is a sizable gap between the actual application needs and the current state of the art. In this paper we make the following contributions.

(a) We present a comprehensive database of labeled videos of faces in challenging, uncontrolled conditions (i.e., ‘in the wild’), the ‘YouTube Faces’ database, along with benchmark, pair matching tests1.

(b) We employ our benchmark to survey and compare the performance of a large variety of existing video face recognition techniques. Finally,

(c) We describe a novel set-to-set similarity measure, the Matched Background Similarity (MBGS). This similarity is shown to considerably improve performance on the benchmark tests.

III.PROPOSED FACE RECOGNITION ALGORITHM

The proposed algorithm is divided into three steps: (i) frame selection, (ii) deep learning based feature extraction, and (iii) face verification using learnt representations. An overview of the proposed algorithm is presented in Fig. 3.

A. Entropy Based Frame Selection

Depending on the frame rate and duration, a video clip of 4−6 seconds may contain 100-200 frames. Existing literature for video face recognition has either used all the frames, or processed some (randomly) selected frames, or have proposed algorithms for frame selection. Processing all the frames can result in inclusion of bad and redundant information. Liu et al. [3] proposed to partition the video into frame clusters and select the most representative frames from each cluster using Principal Component Analysis (PCA). Park et al. [4] proposed to select frames by estimating pose and motion blur information for each frame using Active Appearance Models (AAM) and selecting frames with controlled pose and minimal blur. Jillela and Ross [5] utilized optical flow to create super-resolved frames by using short five frame subsequences while avoiding the sub-sequences which demonstrate high inter-frame motion.

The proposed algorithm presents a novel perspective towards frame selection by utilizing feature richness as the criteria. It is our assertion that quantifying the feature richness of an image helps in extracting the frames that have higher possibility of containing discriminatory features. In order to compute feature-richness, first the input (detected face) image I is preprocessed to a standard size and converted to grayscale. By performing face detection first and considering only the facial region, we ensure that other non-face content of the frame does not interfere with the proposed algorithm. The image is normalized using its mean and standard deviation. Thereafter, the multi-wavelet transform of the

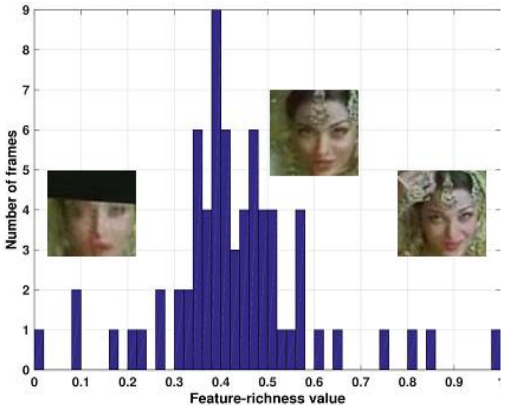


Fig3: Some of the most feature-rich (values close to 1) and least feature-rich frames (values close to 0) are presented for illustration.

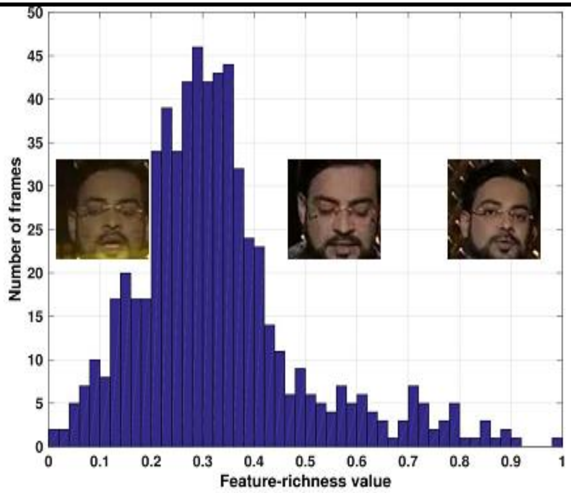


Fig4: Feature-richness distributions for two different videos.

Preprocessed image I is computed as follows:

$$[I_{AP}, I_{HO}\ I_{Vr}, I_{Dg}] = GHM(I) \quad (1)$$

Here, IAp captures the approximation coefficients of the image, whereas [IHo, IV r, IDg] contain the detail coefficients in horizontal, vertical, and diagonal sub-bands respectively.

We use multi wavelet transforms liken GHM (Geronimo, Hardin, and Massopust), Chui and Lian (CL). Multi wavelets are defined using several wavelets with several scaling functions. Multi-wavelets have several advantages in comparison with scalar wavelet. The features such as compact support, Orthogonality, symmetry, and high order approximation are known to be important in signal processing. A scalar wavelet can not possess all these properties at the same time. On the other hand, a multi wavelet system can simultaneously provide perfect reconstruction while preserving length (Orthogonality), good performance at the boundaries (via linear-phase symmetry), and a high order of approximation (vanishing moments). Thus multi wavelets offer the possibility of superior performance and high degree of freedom for image processing applications, compared with scalar wavelets. The detail and approximation coefficients obtained using Eq. 1 represent the first level GHM coefficients. Another level of GHM is applied on the approximation band, IAp, as follows:

$$[I_{AP}, I_{HO}\ I_{Vr}, I_{Dg}] = GHM(I_{AP}); \quad (2)$$

Here, I_{Ap} and [I_{Ho} , I_{Vr} , I_{Dg}] represent the second level GHM approximation and detail coefficients of input image I respectively. GHM is useful to enable multi-resolution analysis of the given image. While the first level GHM presents the coefficients for the finer details of the image, the second level GHM encodes the global features while focusing less on fine details.

We have observed that with images of size 80 × 100. Therefore, in this research, we consider only two levels of GHM. For an image region, entropy signifies the variation in pixel intensity values. To quantify the feature-richness of an image, entropy [9] is computed by using both levels of GHM coefficients. The local entropy of each DWT band is computed by dividing each band into 3 × 3 windows. On applying the algorithm to a GHM band instead of the image, the entropy value captures the local variations in high frequency and approximation sub bands contained in the image. The entropy, H(κ), of an image window κ is computed.

$$H(k) = -\sum_{i=1}^{n} P(Ki)\log2 P(Ki) \quad (3)$$

Where, n is the total number of pixel values, and p(κi) is the value of the probability mass function for κi which represents the probability of pixel value κi appearing in the neighborhood. If the size of the window κ is Mκ × Nκ then

$$P(Ki) = \frac{n(Ki)}{M_k \times N_k} \quad (4)$$

Here, nκi denotes the number of pixels in the window with value κi. The entropy value of each window is combined to compute the feature-richness value of a band.

IJCRT

$$H(F) = \sum_{i=1}^{\omega} (|H(i)|) \quad (5)$$

Here, H F denotes the feature-richness score of a GHM band, ω is the number of windows in the band and Hi denotes the entropy of the i th window. The final score of image I, HF(I), is obtained by aggregating the feature-richness values of individual bands.

$$HF(I) = HF(I'_{AP}) + HF(I'_{HO}) + HF(I'_{Vr}) + HF(I'_{Dg}) + HF(I_{HO}) + HF(I_{Vr}) + HF(I_{Dg}) \quad (6)$$

Given a video V, the feature-richness score of a frame fi is represented as H F(fi). Since the score of each frame depends on the distribution of intensity values in a frame, it is important to normalize the scores across the frames in one video. Let mi represent the feature-richness value corresponding to the i th frame fi, it is obtained using min-max normalization.

$$mi = \frac{HF(fi) - \min(HF)}{\max(HF) - \min(HF)} \quad (7)$$

Where, HF denotes all the feature-richness scores for the video V and min(HF) and max(HF) denote the minimum and maximum values in HF, respectively. Higher values of m signify a more feature-rich frame. Fig. 4 shows the feature richness distribution for two videos of different individuals from the YouTube Faces database [3] along with sample frames of high, average, and low feature-richness values. Once the score of each frame is computed, adaptive frame selection is performed to determine the optimum set of frames to represent a video. Let σ m denote the standard deviation and μm denote the mean pertaining to the set of feature-richness values of the video V. In order to decide which frames are selected for verification, ϕi is computed for each frame

$$\varphi i = \left\{ \begin{array}{ll} 1 & if\ mi \geq \mu m + \frac{\sigma m}{2} \\ 0 & otherwise \end{array} \right\} \quad (8)$$

To perform adaptive frame selection, each frame with ϕ = 1 is selected from a given video. These frames are utilized for feature extraction using the deep learning architecture described in the next section.

B. *Deep Learning Framework for Feature Extraction*

Once the feature-rich frames are obtained, the next step involves feature extraction and matching. Several state-of-theart algorithms in recent literature use convolutional neural networks. In this paper, we propose a stacked denoising auto encoders (SDAE) and Deep Boltzmann Machine (DBM) based algorithm that can yield good results with limited training data while simultaneously being able to utilize additional training data to further improve performance. First, we briefly present an overview of SDAE and DBM followed by the proposed architecture.

1) Stacked Denoising Autoencoder and Deep Boltzmann Machines:

An autoencoder [6], [7] maps the data x ∈ Rα into feature (latent representation) f using a deterministic (encoder) function g such that,

$$f = g\theta(X) = s(W.X + \Delta) \quad (9)$$

is the parameter set, s represents the sigmoid, w is the α × α weight matrix, and is the offset vector of size α. Feature f can be mapped to feature vector x ˆ of dimensionality α using a decoder function g such that,

$$\hat{X} = g'\theta'^{(f)} = s(W'.f + \Delta') \quad (10)$$

Here, = {w, } is the decoder parameter set such that $arg^{min} \|X - \hat{X}\|_2^2$

The parameters are optimized by utilizing the unsupervised training data. Denoising auto encoder [37], a variant of auto encoder, operates on the noisy input data x n and attempts to reconstruct x ˆ such that f = g(x ˆn) = s(w • xn +). It is observed that this variant is robust to noisy data and has good generalizability. Further, adding sparsity constraint helps in learning useful features and the cost function is updated as,

$$\|X - \hat{X}\|_2^2 + \beta \sum_j KL(\rho \parallel \hat{\rho j}) \quad (11)$$

where, ρ is the sparsity parameter, ρ ˆ j is the average activation of the j th hidden unit, K L(ρ ˆ ρ j) = ρ log ρ ρ ˆ j + (1 − ρ) log 1 1− ˆ −ρ ρj is the K L-divergence, and β is the sparsity penalty term. K L divergence measures the difference between a true probability distribution and its approximation. By setting the value of ρ to a small value (such as 0.05), the number of data points for which the j th unit is activated can be forced to be low, which introduces sparsity of features. Smaller values of ρ and larger values of β promote more sparse features.

However, a higher value of β conversely reduces the importance of accurate reconstruction. The values of ρ and β are learnt during the training and validation stages to achieve a tradeoff between reconstruction performance and learning more generalizable features. If the auto encoders are stacked in a layered manner, they are called as stacked auto encoders and form a deep learning architecture to discover “patterns" in the input data.

Deep Boltzmann Machine is an undirected graphical model, deep network architecture, with symmetrically coupled binary units [8]. It is designed by layer-wise training of Restricted Boltzmann Machine (RBM) and stacking them together in an undirected manner. A RBM has stochastic visible and hidden variables which are connected and the energy function is defined as:

$$E(v,h;\theta) = -\sum_{i=1}^{D}\sum_{j=1}^{F} Wij\ vij\ hj - \sum_{i=1}^{D} bivi - \sum_{j=1}^{F} aj\ hj \quad (12)$$

Here, v ∈ {0, 1}D denotes the visible variables and h ∈ {0, 1}F denotes the hidden variables, respectively. The model parameters are denoted by θ = {a, b, W}. Wij denotes the weight of the connection between the i th visible unit and j th hidden unit and bi and a j denote the bias terms of the model. For real valued visible variables such as image pixel intensities, generally, Gaussian-Bernoulli RBMs are utilized and the energy is defined as:

$$E(v,h;\theta) = -\sum_{i=1}^{D}\frac{vi}{\sigma i}\sum_{j=1}^{F} Wij\ \ hj - \sum_{i=1}^{D}\frac{(vi-bi)^2}{2\sigma^2} - \sum_{j=1}^{F} aj\ hj \quad (13)$$

Here, v ∈ RD denotes the real-valued visible vector and θ = {a, b, W, σ} are the model parameters. A single Gaussian Bernoulli RBM can learn a representation of the input data. However, multiple such RBMs can be stacked in a layer wise manner to learn increasingly complex representations of data in the form of a DBM. In this research, a three layer DBM is utilized with a greedy learning approach [9].

A three layer DBM, comprised of Gaussian-Bernoulli RBMs, can learn complex representations of a real-valued input vector v ∈ RD using a sequence of layers of hidden units h(1), h(2), and h(3). The first layer connects the visible units to the first layer of hidden units. Thereafter, subsequent layers connect the hidden units of one layer to the hidden units of the other, causing the hidden units of a layer to act as the visible units for the next layer and so on. The energy of this DBM can be defined as:

$$E(v,h;\theta) = -\sum_{i=1}^{D}\sum_{j=1}^{F1} Wij^{(1)}\frac{vi}{\sigma i}\ hj^{(1)} - \sum_{j=1}^{F1}\sum_{l=1}^{F2} Wij^{(2)}hj^{(1)}hl^{(2)} - \sum_{l=1}^{F2}\sum_{m=1}^{F3} Wim^{(3)}hl^{(2)}hm^{(3)} - \sum_{i=1}^{D}\frac{(vi-bi)^2}{2\sigma^2} - \sum_{j=1}^{F1} aj^{(1)}hj^{(1)} - \sum_{l=1}^{F2} al^{(2)}hl^{(2)} - \sum_{m=1}^{F3} am^{(3)}hm^{(3)} \quad (14)$$

Here, D, F1, F2, F3 are the number of units and visible and hidden layers, and θ = {W(1), W(2), W(3), b, a(1), a(2), a(3), σ} is the set of model parameters representing visible-to-hidden and hidden-to-hidden symmetric connection weights, bias terms, and the Gaussian distribution standard deviation, respectively. The probability assigned by this model to a visible vector v is given by the Boltzmann distribution:

$$P(V;\theta) = \frac{1}{Z(\theta)}\sum_{h}\exp(-E(V,h^{(1)},h^{(2)},h^{(3)};\theta)) \ (15)$$

Here, Z(θ) is the normalizing constant. If only W(1) is considered, the derivative of the log-likelihood with respect to the model parameters is:

$$\frac{\delta \log P(V;\theta)}{\delta W(1)} = Epdata\left[Vh^{(1)^T}\right] - Epdata\left[Vh^{(1)^T}\right] \quad (16)$$

Here, EP data[•] denotes the expectation with respect to the data distribution and EPmodel[•] is the expectation with respect to the distribution defined by the DBM as in Eq. (15). Similar derivatives are obtained for W(1) and W(2), with the product vh(1) replaced by h(1)h(2) and h(2)h(3) respectively.

2) Unsupervised Joint Feature Learning:

SDAE and DBM both individually learn the useful (intermediate) representation of input data. While the SDAE learns two layers of image-level features that can be best utilized to reconstruct the original input, in this paper, we propose a joint representation layer that learns the important features from each constituent layer. This joint layer representation combines two different levels of granularities in features to obtain a better representation. Further, this joint feature is used as input to a DBM to obtain the final representation. While SDAE and joint representation are robust to noise in the input data, DBM learns the internal complex representations probabilistically.

Therefore, it is our assertion that the proposed architecture should be able to produce a robust representation compared to using SDAE or DBM in isolation. Further, DBM is able to interpret the features learned by the joint representation and combine each of its components as required to obtain an enhanced higher level discriminative representation, especially after fine-tuning. Let the size of the input data be M × N; in the proposed architecture, each layer of SDAE is one-fourth the size of its previous layer. Layer-by-layer greedy approach [4] with stochastic gradient descent is utilized to train the SDAE followed by fine-tuning with back-propagation method. Intermediate representations obtained using the 2-hidden layer SDAE are further combined to obtain a joint representation as illustrated in Fig. 5.

The two layers of size M 2 × N 2 and M 4 × N 4 are utilized as input and one joint layer of size 2 × M 4 × N 4 is learned. Let f1 be the representation learned by the first layer of SDAE and f2 be the feature learned by the second layer of SDAE, the joint representation J can be learned using Eq. (17).

$$J = G(f1, f2) \quad (17)$$

Here, G is the joint learning function to obtain J. In this research, using encoder-decoder approach, we define the cost function as:

$$arg_{\Phi}^{min}\left(\|f1 - f1'\|_2^2 + \|f2 - f2'\|_2^2 + R\right) \quad (18)$$

Where, represents the set of all the variables to be learned and R is a regularizer. For ease of explanation, we first

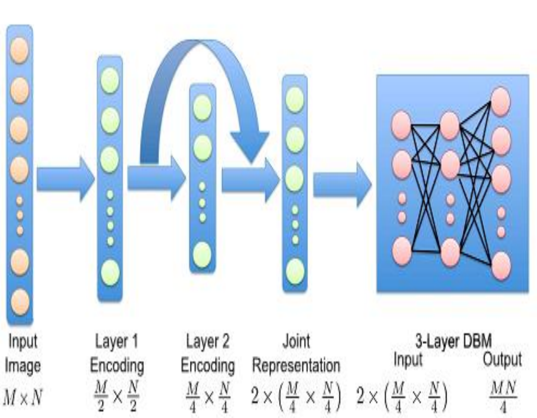

Input
Image
$M \times N$
Layer 1
Encoding
$\frac{M}{2} \times \frac{N}{2}$
Layer 2
Encoding
$\frac{M}{4} \times \frac{N}{4}$
Joint
Representation
$2 \times \left(\frac{M}{4} \times \frac{N}{4}\right)$
3-Layer DBM
Input
$2 \times \left(\frac{M}{4} \times \frac{N}{4}\right)$
Output
$\frac{MN}{4}$

Fig5: Proposed deep learning architecture for facial representation: from input layer (image), two hidden layer representations are computed using SDAE encoding function.

A joint representation is then obtained which combines the information from two SDAE encoding layers. Using joint representation as input, a DBM is used for computing a final feature vector.

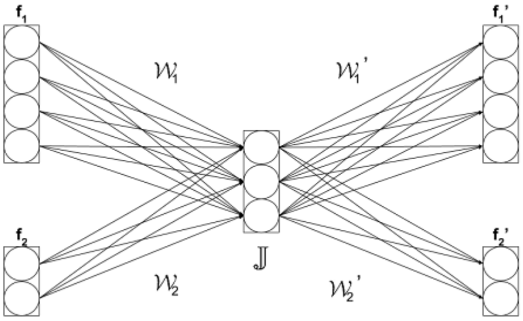


Fig6: Joint learning framework: features learned from the first and second levels of auto encoder, i.e., f1 and f2 are given as input to DBM to learn the joint representation J.

Present the formulation with linear activation. Eq. (17) can be written as,

$$J = W1f1 + W2f2 \quad (19)$$

Using Eq. (18), the associated cost can be written as,

$$arg_{\Phi}^{min}\left(\|f1 - W1'W1f1 - W1'W2f2\|_2^2 + \|f2 - W2'W2f2 - W2'W1f1\|_2^2 + R\right) \quad (20)$$

As shown in Fig. 6, this approach learns the weights = {W1, W2, W1 , W2 } to obtain the joint representation J. In a similar fashion, non-linear cost function can be written as (for simplicity, bias terms are omitted)

$$arg_{\Phi}^{min}\left(\|f1 - s(W1'[s(W1f1)]) - s(W1'[s(W2f2)])\|_2^2 + \|f2 - s(W2'[s(W2f2)]) - s(W2'[s(W1f1)])\|_2^2 + R\right) \quad (21)$$

Adding 2-norm regularization term on W1, W2 and dropout [41] on the joint representation network, Eq. (21) can be written as,

$$arg_{\Phi}^{min}\Big(\|f1 - s(W1'[s(W1f1)]) - s(W1'[s(W2f2)])\|_2^2 + \|f2 - s(W2'[s(W2f2)]) - s(W2'[s(W1f1)])\|_2^2 \\ + (\lambda1\|W1\|_2^2 + \lambda2\|W2\|_2^2)\Big)deopout \quad (22)$$

The joint representation combines abstract and low-level features obtained from SDAE encoding layers and is used as input to a three hidden layer DBM, i.e. J acts as the visible vector. Similar to Eq. (14), the energy of this DBM is represented as:

$$E(J,h;\theta) = -\sum_{i=1}^{D}\sum_{j=1}^{F1} Wij^{(1)}\frac{Ji}{\sigma i}\ hj^{(1)} - \sum_{j=1}^{F1}\sum_{l=1}^{F2} Wjl^{(2)}hj^{(1)}hl^{(2)} - \sum_{l=1}^{F2}\sum_{m=1}^{F3} Wlm^{(3)}hl^{(2)}hm^{(3)} - \sum_{i=1}^{D}\frac{(Ji-bi)^2}{2\sigma^2} - \sum_{j=1}^{F1} aj^{(1)}hj^{(1)} \\ - \sum_{l=1}^{F2} al^{(2)}hl^{(2)} - \sum_{m=1}^{F3} am^{(3)}hm^{(3)} \quad (23)$$

Inspired from [4] and [3], we believe that the learned weight matrix can be modeled as sparse and low rank at the same time and therefore, a regularization approach incorporating both of these can improve feature learning. Hence, we extend the loss function of DBM (RBM) by introducing trace norm regularization technique. Let L be the loss function of RBM (DBM) with the energy function defined in Eq. (23). Along with 1-norm, trace-norm is added to the loss function as follows:

$$L_{new} = L + A\|W\|1 + B\ \|W\|\tau \quad (24)$$

Where • 1 is the 1-norm, and • τ is the trace-norm, and A, B are the regularization parameters which control sparsity and low-rankness. In general, elastic net regularization (• 1 + • 2) [4] may be used; however in this formulation, we propose to utilize trace-norm in conjunction with 1-norm for learning representation in RBM (DBM).

While 1-norm induces sparsity in the weight matrix, trace norm induces features to have low-rankness. The weight matrix learned by the updated loss function has the benefits of both the regularizations and as shown in experimental results, improves the overall verification performance. The size of the first two layers of the DBM is set to 2× M 4 × N 4 and the final layer is set to M N 4 .

A pre-training approach [9] combined with generative fine-tuning [5] is followed to train the DBM. The final hidden layer provides a complex representation of the input which can be utilized for classification.

C. Face Verification Using Feature Richness and Deep Learning Based Representation

As shown in Fig. 3, the proposed framework utilizes the frame selection, feature extraction, and classification architecture for video based face recognition. During training, the stack of SDAE joint representation and DBM is utilized for facial representation. Let I gallery and I probe be the two detected, preprocessed and geometrically normalized face images to be matched. These images are resized to M × N (in our experiments, it is 80 × 100) and converted into vector form.

The trained architecture is used to extract the features from I gallery and I probe, respectively. According to the previous discussion, the input to the feature extraction module is the M N size image vector and the output is a vector of length M N 4 . Features are extracted for each selected frame in a video and given as input to a five layer neural network (one input layer - 3 hidden layers - one output layer) for classification (verification). The neural network classifier is trained to match features extracted from a pair of input images (frames), using all the frames in the training videos. The output of the network is a scalar match score. During testing, the most feature-rich frames are selected from each of the gallery and probe videos, and matched using the proposed feature extraction and matching algorithm. The output of neural network (classifier) is undecimated and match scores are computed.

The videos to be matched may have significant variations in quality and feature-richness. It has been shown in literature that if the images are of very different quality, then the matching performance may deteriorate [46]. Therefore, we perform a post-processing step to select framepairs with similar feature-richness and discard the remaining pairs. Let V1 and V2 be the two videos to be matched, a pair-wise feature-richness value is computed for each possible frame-pair using the algorithm explained in Section II-A.

$$\begin{bmatrix} m1{,}1m1{,}2;\, m2{,}1m2{,}2; \ldots, mi, 1mj, 2; \\ \ldots, mN1{,}1mN2{,}2 \end{bmatrix} \quad (25)$$

mi,1m j,2 denotes the product of feature-richness value associated with the pair formed by the i th frame from V1 and the j th frame from V2. N1 and N2 denote the total number of selected frames from V1 and V2 respectively. Let σm be the standard deviation and μm be the mean pertaining to the set of the pair-wise feature-richness values for all pairs possible between V1 and V2. To finally select the pairs for decision making, following equation is utilized:

$$\gamma_{i,j} = \begin{cases} 1 & if\ mi, 1mj, 2 \geq \mu'm + \dfrac{\sigma'm}{2} \\ 0 & otherwise \end{cases} \quad (26)$$

If the combined score of a pair fi,1 f j,2 is more than the threshold, i.e., if ϒi, j = 1, then this pair is considered for computing the match score. While pairs with ϒi, j < 1 are not considered for verification, other selected frame-pairs are weighted according to the joint feature-richness value. For frame-pair fi,1 f j,2, this weight is computed as ϒi, j mi,1m j,2. A pair where both participating frames are highly featuring rich is assigned a higher weight compared to other combinations. Here, facial coordinates obtained during face detection are used to ensure that frontal-only and semi-profile images are not matched with profile faces (i.e, when pose variations are very large).

The final match score is computed in the form of a weighted sum of scores obtained from each participating frame-pair.

The undecimated/unthresholded network (classifier) output of these pairs are combined using weighted sum rule [28] and a verification threshold is applied to provide the final decision of accept or reject (same or not same) at a fixed false accept rate.

IV.RESULTS

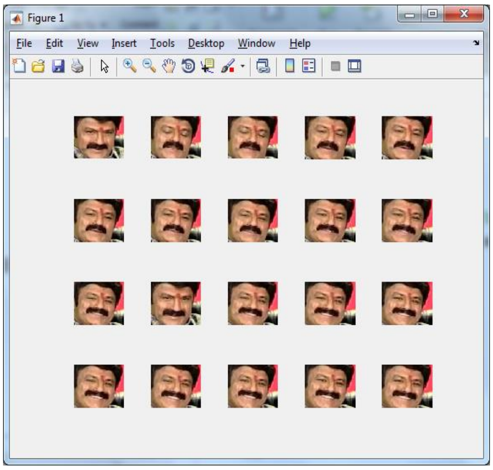


Fig7: Video face images for training the data

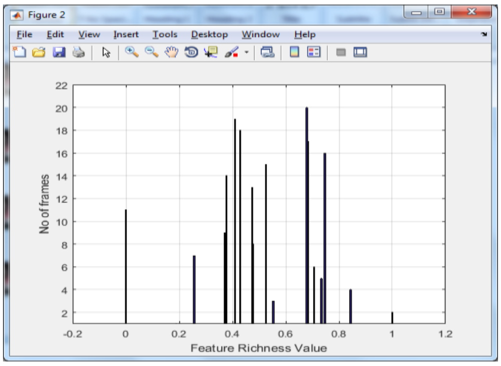
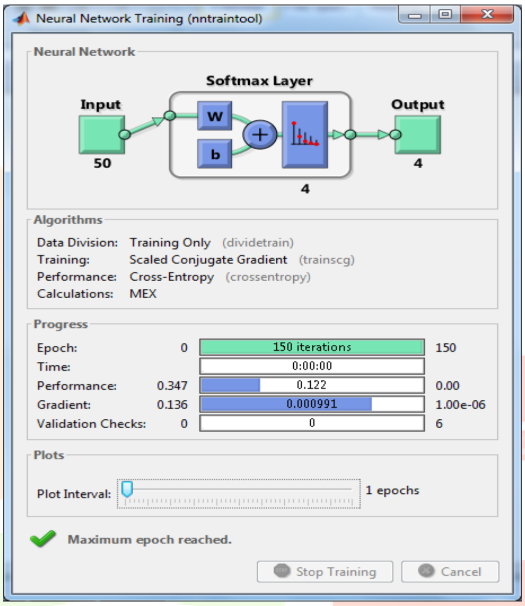


Fig8: Feature reachness value


Neural Network Training (nntraintool)

Neural Network

Algorithms

Data Division: Training Only (dividetrain)
Training: Scaled Conjugate Gradient (trainscg)
Performance: Cross-Entropy (crossentropy)
Calculations: MEX

Progress

Epoch:	0	150 iterations	150
Time:		0:00:00	
Performance:	0.347	0.122	0.00
Gradient:	0.136	0.000991	1.00e-06
Validation Checks:	0	0	6

Plots

Plot Interval: 1 epochs

✓ Maximum epoch reached.

Fig.9: NN training

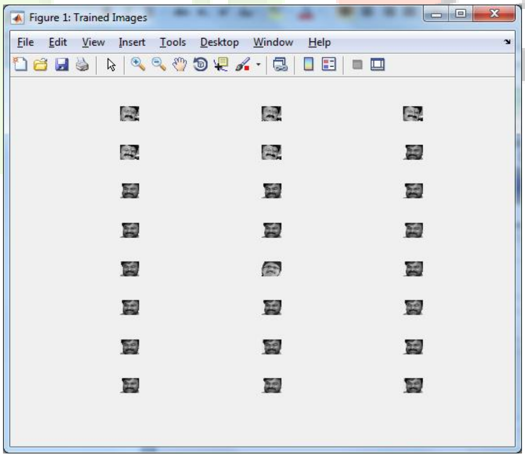


Fig.10: Training images for classifier

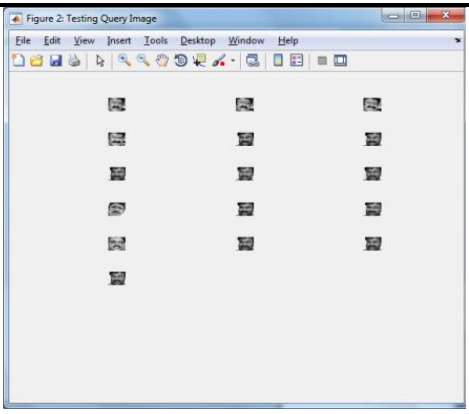


Figure 2: Testing Query Image
File Edit View Insert Tools Desktop Window Help

Fig.11: Testing images for classifier

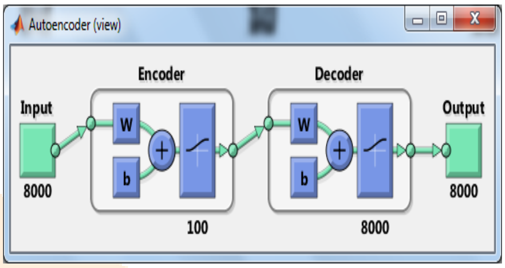


Fig.12: Autoencoder

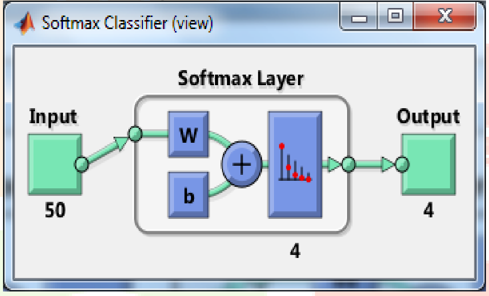


Fig.13: Classifier View

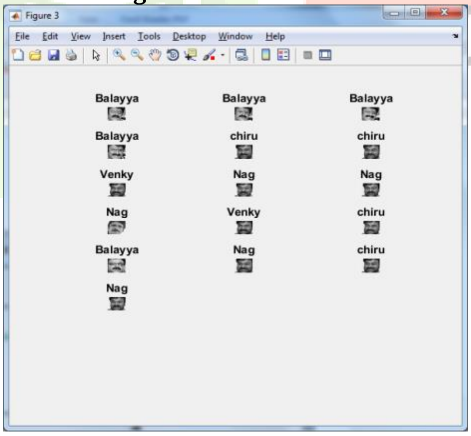


Fig.14: Face verified results

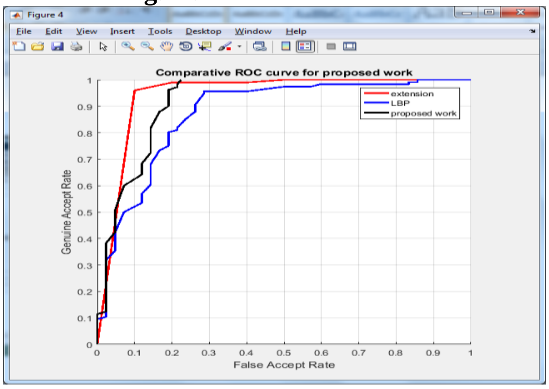


Fig.15: ROC graph for extension comparison

IV.CONCLUSION

The proposed algorithm starts off evolved with adaptively choosing feature-wealthy frames from enter films the use of wavelet decomposition and entropy. The proposed deep gaining knowledge of structure which combines SDAE joint illustration with DBM is used to extract capabilities from the selected frames. The extracted representations from two films are matched the use of a feed ahead neural network. The outcome is validated at the YouTube Faces databases. The evaluation with modern-day consequences on both the databases show that the proposed algorithm offers the exceptional results on both the databases at low false receive price, even with constrained schooling records.

REFERENCES

[1] Facial recognition technology safeguards Beijing Olympics, accessed on Mar. 10, 2017 [Online].Available:http://english.cas.cn/resources/archive/china_archive/cn2008/200909/t20090923_42959.shtml]

[2] J. Beveridge et al., "The challenge of face recognition from digital pointand-shoot cameras," in Proc. IEEE Conf. Biometrics Theory, Appl. Syst., Oct. 2013, pp. 1–8.

[3] L. Wolf, T. Hassner, and I. Maoz, "Face recognition in unconstrained videos with matched background similarity," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2011, pp. 529–534.

[4] L. Wolf and N. Levy, "The SVM-minus similarity score for video face recognition," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2013, pp. 3523–3530.

[5] H. Li, G. Hua, Z. Lin, J. Brandt, and J. Yang, "Probabilistic elastic matching for pose variant face verification," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2013, pp. 3499–3506.

[6] Z. Cui, W. Li, D. Xu, S. Shan, and X. Chen, "Fusing robust face region descriptors via multiple metric learning for face recognition in the wild," in Proc.

[7] H. Méndez-Vázquez, Y. Martínez-Díaz, and Z. Chai, "Volume structured ordinal features with background similarity measure for video face recognition," in Proc. Int. Conf. Biometrics (ICB), Jun. 2013, pp. 1–6.

[8] H. S. Bhatt, R. Singh, and M. Vatsa, "On recognizing faces in videos using clustering-based re-ranking and fusion," IEEE Trans. Inf. Forensics Security, vol. 9, no. 7, pp. 1056–1068, Jul. 2014.

[9] J. Y. Junlin Hu, J. Lu, and Y.-P. Tan, "Large margin multi-metric learning for face and kinship verification in the wild," in Proc. Asian Conf. Comput. Vis., 2014, pp. 252–267.

[10] J. Hu, J. Lu, and Y. Tan, "Discriminative deep metric learning for face verification in the wild," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2014, pp. 1875–1882.

[11] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "DeepFace: Closing the gap to human-level performance in face verification," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2014, pp. 1701–1708.