# UNCONSTRAINED FACE VERIFICATION USING DEEP CNN FEATURES

Pallekonda Pavani [1], M.Revathi [2], S.RAVI KUMAR [3]
[1]PG Scholar, ECE Department, PVKK Institute of Technology, Anantapur
[2] Asst Professor, ECE Department, PVKK Institute of Technology, Anantapur
[3] Assoc Professor ,ECE Department, PVKK Institute of Technology , Anantapur

Abstract: Abundance and availability of video capture devices, such as mobile phones and surveillance cameras, have instigated research in video face recognition, which is highly pertinent in law enforcement applications. While the current approaches have reported high accuracies at equal error rates, performance at lower false accept rates requires significant improvement. In this project, we propose a novel face verification algorithm, which starts with selecting feature-rich frames from a video sequence using discrete wavelet transform and entropy computation. Frame selection is followed by representation learning-based feature extraction, where three contributions are presented, deep learning architecture, which is a combination of stacked denoising sparse autoencoder (SDAE) and deep Boltzmann machine (DBM); formulation for joint representation in an autoencoder; and 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 demonstrated on two publicly available databases, YouTube Faces and Point and Shoot Challenge. Experimental analysis suggests that, the proposed feature richness based frame selection offers noticeable and consistent performance improvement compared with frontal only frames, random frames, or frame selection using perceptual no-reference image quality measures and the joint feature learning in SDAE and sparse and low rank regularization in DBM helps in improving face verification performance. On the benchmark Point and Shoot Challenge database, the algorithm yields the verification accuracy of over 97% at 1% false accept rate whereas, on the YouTube Faces database, over 95% verification accuracy is observed at equal error rate.

## 1. INTRODUCTION

Face recognition is one of the major issues in biometric technology. It identifies and/or verifies a person by using 2D/3D physical characteristics of the face images. The baseline method of face recognition system is the eigenface by which the goal of the eigenface method is to project linearly the image space onto the feature space which has less dimensionality. One can reconstruct a face image by using only a few eigenvectors which correspond to the largest eigenvalues, known as eigenpicture, eigenface, Karhunen- Loeve transform and principal component analysis ,Several techniques have been proposed for solving a major problem in face recognition such as fisher face , elastic bunch graph matching and support vector machine. However, there are still many challenge problems in face recognition system such as facial expressions, pose variations, occlusion and illumination change. Those variations dramatically degrade the performance of face recognition system. It is evident that illumination variation is the most impact of the changes in appearance of the face images because of its fluctuation by increasing or decreasing the intensities of face images due to shadow cast given by different light source direction. Therefore the one of key success is to increase the robustness of face representation against these variations.

In order to reduce the illumination variation, many literatures have been proposed. Belhumeur etc. All suggested that discarding the three most significant principal components can reduce the illumination variation in the face images. Nevertheless, the three most significant principal components not only contain illumination variations but also some useful information, therefore, the system was also degraded as well. Wang et. al. proposed a Self Quotient Image (SQI) by using only single image. The 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 The normalized image was obtained by dividing the original image with the large scale one. The TVQI and LTV has a very high computational complexity due to the second order cone programming as their kernel function.

However these methods are suitable only for illumination variation but not for other variations. Whereas the face representation based method has more robustness. It is not insensitive to illumination variation but insensitive to facial expression as well, such as Local Binary Pattern (LBP) and its extension was originally designed for texture description. The LBP operator assigns a label to every pixel of an image by thresholding the 3x3-neighbourhood of each surrounding pixel with the center pixel value and a decimal representation is then obtained from the binary sequence (8 bits). The LBP image is subsequently divided into *R* nonoverlapping regions of same size and the local histogram over each regions are then calculated. Finally the concatenated histogram can be obtained as a face descriptor.

## 2. EXISTING SYSTEM

In existing Different types of method are implemented. Face recognition (principal component analysis (PCA), LDA, ICA, and SVMs) to assess the feasibility of real world face recognition. One of the important technique of recognition is template matching in which a template to recognize is available and is compared with already stored template. In our approach PCA method for feature extraction and matching is used. Principal Component Analysis: PCA is used to reduce the dimensionality of the image while preserving much of the information. It is the powerful tool for analyzing the data by identifying patterns in the dataset and reduces the dimensions of the dataset such that maximum variance in the original data is visible in reduced data.

PCA was invented by Karl Pearson in 1901. It works by converting set of correlated variables to linearly uncorrelated variable called principal components. Principal components are calculated by computing Eigen vectors of covariance matrix obtained from the group of hand images. The highest M eigenvectors contains the maximum variance in the original data. These principal components are orthogonal to each other and the first component is in the direction of greatest variance.

We can use PCA to compute and study the Eigenvectors of the different pictures and then to express each image with its principal components (Eigenvectors). It is a way of identifying patterns in data, and expressing the data in such a way as to highlight their similarities and differences. First of all, we had to create the data set. The aim is to choose a good number of pictures and a good resolution of these in order to have the best recognition with the smallest database. Then, the next step is to subtract the mean from each of the data dimensions. The mean subtracted is simply the average across each dimension. The step three is to calculate the covariance matrix of the database. We could not calculate the covariance matrix of the first matrix, because it was too huge. So we had to find a way to find out the principal eigenvectors without calculating the big covariance matrix. The method consists in choosing a new covariance matrix

The PCA approach has 2 stages: Training and Testing stage. In the training stage the Eigen space in established using training images of hand gestures and these images are mapped to the Eigen space. In the testing stage the image to be tested is mapped to same Eigen space and is classified using distance classifier.

## 3. PROPOSED SYSTEM

VIDEO face recognition has become highly significant in surveillance scenarios. For example, more than 80,000 people were identified and verified during the 2008 Beijing Olympics with the help of face recognition in videos . With advancements in technology, video capturing devices are accessible to a large number of people in the form of portable electronic devices such as 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. 5.1 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. Video face recognition has been extensively studied and several algorithms have been proposed. The image provides a review of some of the algorithms along with the summary of results reported on popular video face recognition databases.

**Fig 1: A subset of frames illustrating the amount of information present in a video.**

A single video can capture a subject's face under different pose, expression, and illumination variations. Video face recognition algorithms can broadly be classified into two types: (a) set-based and (b) sequence based . The set-based approaches consider a video as a set of images (frames) which are then modeled and matched using a variety of methodologies. These approaches may not utilize the temporal information contained in the video, i.e. the order of frames in the original video may not matter. On the other hand, sequence-based approaches are specifically designed to utilize temporal information of the video. These approaches model the video as a sequence of images and apply sequence classification techniques for recognition. Some of the recent techniques utilize large image dictionaries to characterize videos, while some others have focused on metric learning based approaches or deep learning based approaches . For comparison, the results are generally reported on benchmark database

**3.1 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. 5.2.

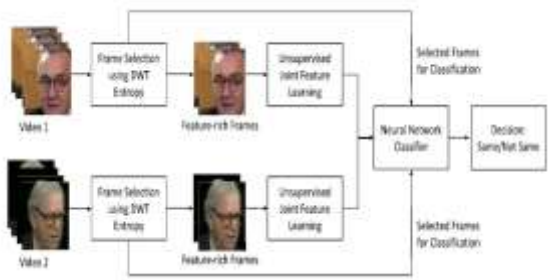

**Fig 2: Illustrating the steps involved in the proposed face recognition algorithm.**

**3.1.1 Frame Selection:**

The proposed method is an Daubechies complex wavelet domain. It uses frame differencing for obtaining video object planes which gives the changed pixel value from consecutive frames. First, we decompose two consecutive frames ($I_{n-1}$ and $I_n$) using complex wavelet domain and then apply approximate median filter based method to detect frame difference

**3.1.2 Frame Substraction Method:**

We decompose two consecutive frames ($I_{n-1}$ and $I_n$) using complex wavelet domain and then apply approximate median filter based method to detect frame difference For every pixellocation (i, j) _ the co-ordinate of frame of frame $I_n$ ($i$ , $j$) and $I_{n-1}(i , j)$ respectively

$$FD_n(i , j) = WI_n(i , j) - WI_{n-1}(i , j) \quad (1)$$

**3.1.3 Discrete Wavelet Transform:**

The foundations of the DWT go back to 1976 when Croiser, Esteban, and Galand devised a technique to decompose discrete time signals. Crochiere, Weber, and Flanagan did a similar work on coding of speech signals in the same year. They named their analysis scheme as **subband coding**. In 1983, Burt defined a technique very similar to subband coding and named it **pyramidal coding** which is also known as multiresolution analysis. Later in 1989, Vetterli and Le Gall made some improvements to the subband coding scheme, removing the existing redundancy in the pyramidal coding scheme. Subband coding is explained below. A detailed coverage of the discrete wavelet transform and theory of multiresolution analysis can be found in a number of articles and books that are available on this topic, and it is beyond the scope of this tutorial.

**3.1.3.1 The Subband Coding and The Multiresolution Analysis :**

The main idea is the same as it is in the CWT. A time-scale representation of a digital signal is obtained using digital filtering techniques. Recall that the CWT is a correlation between a wavelet at different scales and the signal with the scale (or the frequency) being used as a measure of similarity. The continuous wavelet transform was

computed by changing the scale of the analysis window, shifting the window in time, multiplying by the signal, and integrating over all times. In the discrete case, filters of different cutoff frequencies are used to analyze the signal at different scales. The signal is passed through a series of high pass filters to analyze the high frequencies, and it is passed through a series of low pass filters to analyze the low frequencies.

The resolution of the signal, which is a measure of the amount of detail information in the signal, is changed by the filtering operations, and the scale is changed by upsampling and downsampling (subsampling) operations. Subsampling a signal corresponds to reducing the sampling rate, or removing some of the samples of the signal. For example, subsampling by two refers to dropping every other sample of the signal. Subsampling by a factor *n* reduces the number of samples in the signal *n* times.

Upsampling a signal corresponds to increasing the sampling rate of a signal by adding new samples to the signal. For example, upsampling by two refers to adding a new sample, usually a zero or an interpolated value, between every two samples of the signal. Upsampling a signal by a factor of *n* increases the number of samples in the signal by a factor of *n*.

Although it is not the only possible choice, DWT coefficients are usually sampled from the CWT on a dyadic grid, i.e., $s_0 = 2$ and $s_0 = 1$, yielding $s=2^j$ and $s=k*2^j$, as described. Since the signal is a discrete time function, the terms function and sequence will be used interchangeably in the following discussion. This sequence will be denoted by x[n], where n is an integer.The procedure starts with passing this signal (sequence) through a half band digital lowpass filter with impulse response h[n]. Filtering a signal corresponds to the mathematical operation of convolution of the signal with the impulse response of the filter. The convolution operation in discrete time is defined as follows:

$$x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[k].h[n-k]$$

A half band lowpass filter removes all frequencies that are above half of the highest frequency in the signal. For example, if a signal has a maximum of 1000 Hz component, then half band lowpass filtering removes all the frequencies above 500 Hz. The unit of frequency is of particular importance at this time. In discrete signals, frequency is expressed in terms of radians. Accordingly, the sampling frequency of the signal is equal to $2\pi$ radians in terms of radial frequency. Therefore, the highest frequency component that exists in a signal will be $2\pi$ radians, if the signal is sampled at Nyquist's rate (which is twice the maximum frequency that exists in the signal); that is, the Nyquist's rate corresponds to $2\pi$ rad/s in the discrete frequency domain. Therefore using Hz is not appropriate for discrete signals. However, Hz is used whenever it is needed to clarify a discussion, since it is very common to think of frequency in terms of Hz. It should always be remembered that the unit of frequency for discrete time signals is radians.

After passing the signal through a half band lowpass filter, half of the samples can be eliminated according to the Nyquist's rule, since the signal now has a highest frequency of $\pi$ /2 radians instead of $\pi$ radians. Simply discarding every other sample will **subsample** the signal by two, and the signal will then have half the number of points. The scale of the signal is now doubled. Note that the lowpass filtering removes the high frequency information, but leaves the scale unchanged. Only the subsampling process changes the scale. Resolution, on the other hand, is related to the amount of information in the signal, and therefore, it is affected by the filtering operations. Half band lowpass filtering removes half of the frequencies, which can be interpreted as losing half of the information. Therefore, the resolution is halved after the filtering operation. Note, however, the subsampling operation after filtering does not affect the resolution, since removing half of the spectral components from the signal makes half the number of samples redundant anyway. Half the samples can be discarded without any loss of information. In summary, the lowpass filtering halves the resolution, but leaves the scale unchanged. The signal is then subsampled by 2 since half of the number of samples are redundant. This doubles the scale. This procedure can mathematically be expressed as

$$y[n] = \sum_{k=-\infty}^{\infty} h[k].x[2n-k]$$

Having said that, we now look how the DWT is actually computed: The DWT analyzes the signal at different frequency bands with different resolutions by decomposing the signal into a

coarse approximation and detail information. DWT employs two sets of functions, called scaling functions and wavelet functions, which are associated with low pass and highpass filters, respectively. The decomposition of the signal into different frequency bands is simply obtained by successive highpass and lowpass filtering of the time domain signal. The original signal x[n] is first passed through a halfband highpass filter g[n] and a lowpass filter h[n]. After the filtering, half of the samples can be eliminated according to the Nyquist's rule, since the signal now has a highest frequency of $\pi$ /2 radians instead of $\pi$. The signal can therefore be subsampled by 2, simply by discarding every other sample. This constitutes one level of decomposition and can mathematically be expressed as follows:

$$y_{high}[k] = \sum_n x[n] \cdot g[2k-n]$$

$$y_{low}[k] = \sum_n x[n] \cdot h[2k-n]$$

where $y_{high}[k]$ and $y_{low}[k]$ are the outputs of the highpass and lowpass filters, respectively, after subsampling by 2.

This decomposition halves the time resolution since only half the number of samples now characterizes the entire signal. However, this operation doubles the frequency resolution, since the frequency band of the signal now spans only half the previous frequency band, effectively reducing the uncertainty in the frequency by half. The above procedure, which is also known as the subband coding, can be repeated for further decomposition. At every level, the filtering and subsampling will result in half the number of samples (and hence half the time resolution) and half the frequency band spanned (and hence double the frequency resolution). Below figure illustrates this procedure, where x[n] is the original signal to be decomposed, and h[n] and g[n] are lowpass and highpass filters, respectively. The bandwidth of the signal at every level is marked on the figure as "f".

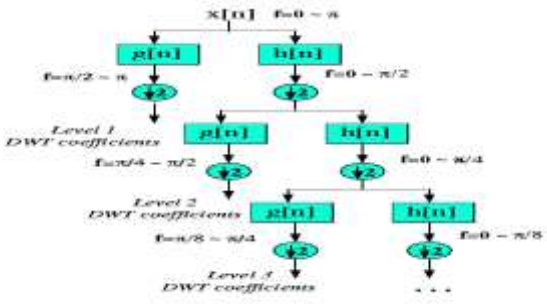


**Fig 3 : The Subband Algorithm Tree Diagram**

The Subband Coding Algorithm As an example, suppose that the original signal x[n] has 512 sample points, spanning a frequency band of zero to $\pi$ rad/s. At the first decomposition level, the signal is passed through the highpass and lowpass filters, followed by subsampling by 2. The output of the highpass filter has 256 points (hence half the time resolution), but it only spans the frequencies $\pi$ /2 to $\pi$ rad/s (hence double the frequency resolution). These 256 samples constitute the first level of DWT coefficients. The output of the lowpass filter also has 256 samples, but it spans the other half of the frequency band, frequencies from 0 to $\pi$ /2 rad/s. This signal is then passed through the same lowpass and highpass filters for further decomposition. The output of the second lowpass filter followed by subsampling has 128 samples spanning a frequency band of 0 to $\pi$/4 rad/s, and the output of the second highpass filter followed by subsampling has 128 samples spanning a frequency band of $\pi$/4 to $\pi$/2 rad/s. The second highpass filtered signal constitutes the second level of DWT coefficients. This signal has half the time resolution, but twice the frequency resolution of the first level signal. In other words, time resolution has decreased by a factor of 4, and frequency resolution has increased by a factor of 4 compared to the original signal. The lowpass filter output is then filtered once again for further decomposition. This process continues until two samples are left. For this specific example there would be 8 levels of decomposition, each having half the number of samples of the previous level. The DWT of the original signal is then obtained by concatenating all coefficients starting from the last level of decomposition (remaining two samples, in this case). The DWT will then have the same number of coefficients as the original signal.

The frequencies that are most prominent in the original signal will appear as high amplitudes in that region of the DWT signal that includes those particular frequencies. The difference of this transform from the Fourier transform is that the time localization of these frequencies will not be lost. However, the time localization will have a resolution that depends on which level they appear. If the main information of the signal lies in the high frequencies, as happens most often, the time localization of these frequencies will be more precise, since they are characterized by more number of samples. If the main information lies only at very low frequencies, the time localization will not be very precise, since few samples are used to express signal at these frequencies. This procedure in effect offers a good time resolution at high frequencies, and good frequency resolution at low frequencies. Most practical signals encountered are of this type. The frequency bands that are not very prominent in the original signal will have very low amplitudes, and that part of the DWT signal can be discarded without any major loss of information, allowing data reduction. We will revisit this example, since it provides important insight to how DWT should be interpreted. Before that, however, we need to conclude our mathematical analysis of the DWT.

One important property of the discrete wavelet transform is the relationship between the impulse responses of the highpass and lowpass filters. The highpass and lowpass filters are not independent of each other, and they are related by

$$g[L-1-n] = (-1)^n \cdot h[n]$$

where g[n] is the highpass, h[n] is the lowpass filter, and L is the filter length (in number of points). Note that the two filters are odd index alternated reversed versions of each other. Lowpass to highpass conversion is provided by the $(-1)^n$ term. Filters satisfying this condition are commonly used in signal processing, and they are known as the Quadrature Mirror Filters (QMF). The two filtering and subsampling operations can be expressed by

$$y_{high}[k] = \sum_n x[n] \cdot g[-n+2k]$$

$$y_{low}[k] = \sum_n x[n] \cdot h[-n+2k]$$

The reconstruction in this case is very easy since halfband filters form orthonormal bases. The above procedure is followed in reverse order for the reconstruction. The signals at every level are upsampled by two, passed through the synthesis filters g'[n], and h'[n] (highpass and lowpass, respectively), and then added. The interesting point here is that the analysis and synthesis filters are identical to each other, except for a time reversal. Therefore, the reconstruction formula becomes (for each layer)

$$x[n] = \sum_{k=-\infty}^{\infty}(y_{high}[k].g[-n+2k]) + (y_{low}[k].h[-n+2k])$$

However, if the filters are not ideal halfband, then perfect reconstruction cannot be achieved. Although it is not possible to realize ideal filters, under certain conditions it is possible to find filters that provide perfect reconstruction. The most famous ones are the ones developed by Ingrid Daubechies, and they are known as Daubechies' wavelets.

Note that due to successive subsampling by 2, the signal length must be a power of 2, or at least a multiple of power of 2, in order this scheme to be efficient. The length of the signal determines the number of levels that the signal can be decomposed to. For example, if the signal length is 1024, ten levels of decomposition are possible. Interpreting the DWT coefficients can sometimes be rather difficult because the way DWT coefficients are presented is rather peculiar. To make a real long story real short, DWT coefficients of each level are concatenated, starting with the last level. An example is in order to make this concept clear:

Suppose we have a 256-sample long signal sampled at 10 MHZ and we wish to obtain its DWT coefficients. Since the signal is sampled at 10 MHz, the highest frequency component that exists in the signal is 5 MHz. At the first level, the signal is passed through the lowpass filter h[n], and the highpass filter g[n], the outputs of which are subsampled by two. The highpass filter output is the first level DWT coefficients. There are 128 of them, and they represent the signal in the [2.5 5] MHz range. These 128 samples are the last 128 samples plotted. The lowpass filter output, which also has 128 samples, but spanning the frequency band of [0 2.5] MHz, are further decomposed by passing them through the same h[n] and g[n]. The output of the second highpass filter is the level 2 DWT coefficients and these 64 samples precede

the 128 level 1 coefficients in the plot. The output of the second lowpass filter is further decomposed, once again by passing it through the filters h[n] and g[n]. The output of the third highpass filter is the level 3 DWT coefficients. These 32 samples precede the level 2 DWT coefficients in the plot. The procedure continues until only 1 DWT coefficient can be computed at level 9. This on coefficient is the first to be plotted in the DWT plot. This is followed by 2 level 8 coefficients, 4 level 7 coefficients, 8 level 6 coefficients, 16 level 5 coefficients, 32 level 4 coefficients, 64 level 3 coefficients, 128 level 2 coefficients and finally 256 level 1 coefficients. Note that less and less number of samples is used at lower frequencies, therefore, the time resolution decreases as frequency decreases, but since the frequency interval also decreases at low frequencies, the frequency resolution increases. Obviously, the first few coefficients would not carry a whole lot of information, simply due to greatly reduced time resolution.

To illustrate this richly bizarre DWT representation let us take a look at a real world signal. Our original signal is a 256-sample long ultrasonic signal, which was sampled at 25 MHz. This signal was originally generated by using a 2.25 MHz transducer, therefore the main spectral component of the signal is at 2.25 MHz. The last 128 samples correspond to [6.25 12.5] MHz range. As seen from the plot, no information is available here, hence these samples can be discarded without any loss of information. The preceding 64 samples represent the signal in the [3.12 6.25] MHz range, which also does not carry any significant information. The little glitches probably correspond to the high frequency noise in the signal. The preceding 32 samples represent the signal in the [1.5 3.1] MHz range. As you can see, the majority of the signal's energy is focused in these 32 samples, as we expected to see. The previous 16 samples correspond to [0.75 1.5] MHz and the peaks that are seen at this level probably represent the lower frequency envelope of the signal. The previous samples probably do not carry any other significant information. It is safe to say that we can get by with the 3rd and 4th level coefficients, that is we can represent this 256 sample long signal with 16+32=48 samples, a significant data reduction which would make your computer quite happy.

One area that has benefited the most from this particular property of the wavelet transforms is image processing. As you may well know, images, particularly high-resolution images, claim a lot of disk space. As a matter of fact, if this tutorial is taking a long time to download, that is mostly because of the images. DWT can be used to reduce the image size without losing much of the resolution. Here is how: For a given imge, you can compute the DWT of, say each row, and discard all values in the DWT that are less then a certain threshold. We then save only those DWT coefficients that are above the threshold for each row, and when we need to reconstruct the original image, we simply pad each row with as many zeros as the number of discarded coefficients, and use the inverse DWT to reconstruct each row of the original image

### 3.2 Deep Learning Framework for Feature Extraction:

First introduced the concept of fusing convolutional and sub sampling layers in SDAE for handwritten digit recognition. The main principle is to convolve with a stride (step size) of 2 or more on an input feature map with a convolution kernel. This is equivalent to a normal convolution operation followed by sub sampling performed in a convolutional layer. The result is a significant performance speedup with the expense of a small overhead in total trainable parameters. The fusion of convolution and sub sampling can be described by the equation:

$$Y_j^{(l)}(x,y) = f\left(\sum_{i=0}^{N}\sum_{u=0}^{K_x^{(l)}}\sum_{v=0}^{K_y^{(l)}} X_i^{(l)} + \theta_j^{(l)}\right)$$

$$X_i^{(l)} = Y_i^{(l-1)}(S_x{}^{(l)}x + u, S_y{}^{(l)}y + v)w_{ji}^{(l)}(u,v)$$

Where f()denotes the activation function,) and are the input and output feature maps, respectively is the convolutional kernel weight is the bias, that represents the total number of input feature map, is the horizontal convolution step size) is the vertical convolution step size and are the width and height of convolutional kernels, respectively. In this work, the activation function applied is the scaled hyperbolic tangent:where A denotes the amplitude of the function, and B determines its slopes at the origin. The values of A and B are chosen to be.

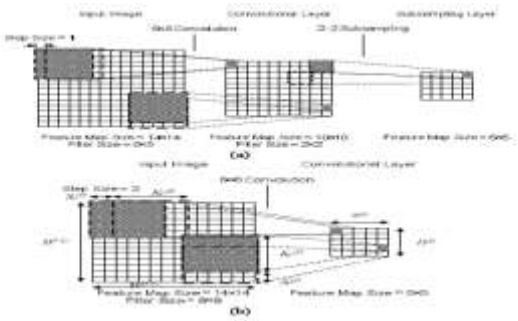

**Fig 4: Convolutional layer (a) Convolution (with stride of 1) followed by subsampling; (b) Convolution operation (with stride of 2)**

depicts the fusion of the convolutional and sub sampling layers. In Fig.5.4(a), a 5×5 convolution is performed on the input image followed by a 2×2 sub sampling operation. The convolution is performed with a stride of 1. In Fig.5.4(b), a convolution with a stride of 2 is shown, which generates a feature map of equivalent size, thus resembling an operation of convolution followed by sub sampling

### 3.3 Unsupervised Joint Feature Learning:

The general equation for a 2D discrete convolution and 2D discrete cross-correlation are given by the following equation

$$Y(x,y) = \sum_{u=0}^{Kx}\sum_{v=0}^{Ky} X(x-u, y-v)w(u,v)$$

$$Y(x,y) = \sum_{u=0}^{Kx}\sum_{v=0}^{Ky} X(x+u, y+v)w(u,v)$$

where is an input image, is the output image is the kernel weight, and represent the width and height of convolutional kernel, respectively. It is clear from these equations that convolution and cross-correlation performs similar mathematical operations, except for the flipping of kernel coefficients in a convolution. This property is usually ignored in image processing tasks, as these filter coefficients are usually symmetrical. However, in a convolution layer, the kernel coefficients (weights) are randomly initialized, hence contributing to different results between a convolution and a cross-correlation operation. Since a flipping operation generates more computations, resulting in higher logic utilization in hardware, we propose to replace the 2D discrete convolution operation with a 2D discrete cross-correlation, hence easing the computational complexity

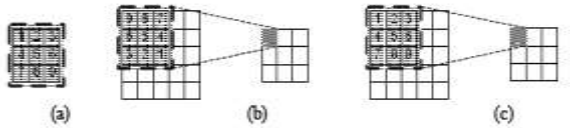


**Fig 5 : 2D discrete convolution (a) Original convolutional kernel, (b) Convolution with flipped kernel, (c) Convolution with original kernel.**

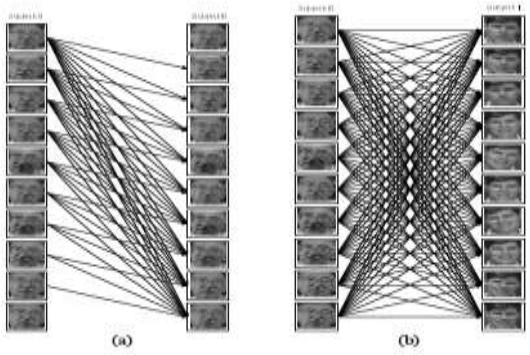


**Fig 6: Generation of image pairs (a) between images of same subject (b) between images of two different subjects**

### 3.4 Energy-Based Learning:

In energy-based models (EBM), unnormalized energy is assigned to every possible configuration of the variables being modeled. Prediction is performed by searching the combination of variables that minimizes the energy. This energy can be associated with a trainable similarity metric, where lower energy corresponds to higher similarity, while distinct dissimilarity is represented by high energy. This relationship can be mathematically expressed as parameters that minimizes a loss function. The loss function consists of two partial loss functions: one to decrease energy of similar pairs, and another to increase the energy if the pair is deemed dissimilar (known as contrastive loss function). This function is given as follows imposter pairs, and represents the pair label (0 for genuine and 1 for imposter). In this paper, we apply the contrastive loss function from, which can effectively discriminate between genuine and imposter pairs. For a simpler computation, L1 norm (Manhattan distance) is chosen as similarity measure instead of L2 norm. The rationale is that the gradient of the energy (square norm) with

respect to parameter would be negligible as the energy is near to zero, and the machine may fail to learn whenever the energy of imposter pair approaches zero The contrastive loss function used to learn the similarity metric is given by the equation $L(W,Y,X_1,X_2) = (1-Y)\,L_G(E_W) + Y\,L_1(E_W) = (1-Y)\frac{2}{\Omega}(E_W) + (Y)2Qe^{\frac{-2.7726E_W}{Q}}$

corresponds to the energy level. Training using this loss function decreases the energy level for a matched pair, while increasing the energy level when face images of two distinctive (different) subjects are presented to the network. Ideally, all genuine pairs will produce low energy levels, while high energy levels correspond to imposter pairs. We randomly initialize the network weights and biases based on Gaussian distribution, with mean value of 0 and standard deviation of 0.05. The proposed Siamese SDAE system is trained using stochastic first order gradient descent method with annealed learning rate based on number of training epochs. The annealed global learning rate is defined by the equation

$$\epsilon^{t+1} = \begin{cases} \epsilon_{max} & t = 0 \\ \epsilon_{min} & \epsilon^t < \epsilon_{min} \\ \epsilon^t \times \alpha & otherwise \end{cases}$$

where denotes the fading factor of global learning rate maximum (initial) global learning rate, and is minimum global learning rate. In this paper, we set the fading factor. The learning rate value is updated after every training epoch. No other learning parameters such as momentum and weight decay are incorporated into the learning process, hence reducing the total parameters that need to be tuned.

**5.5 Cross Database Experiments**:

The generalizability of an algorithm can be evaluated in situations where the training and testing data belong to different databases, i.e, cross-database experiments. To evaluate the effectiveness of the proposed algorithm in cross database scenarios, we have performed three different experiments:

- Training and testing databases belong to the same database. For instance, training with YouTube faces train set and testing with YouTube faces test set.
- Training and testing databases belong to different databases. For instance, training with YouTube faces train set and testing with PaSC test set.
- Training database is from multiple databases whereas, the testing is performed with a single database. For instance, training with both YouTube faces and PaSC train sets and testing on YouTube faces test set.

**RESULT**

**Input video :**

Here we are going to give the input video which is in .avi format. Here we are taking the input video which consists of two persons and we are going to find out whether the persons are authorized are not. In this we will select the input video which is stores in the code folder. In the further process we are going to do the sampling process by using discrete wavelet transform in 2 levels where the exact sampling will be done to clearly check the difference between to persons that are there in the input video. The input video has 15 frames per second and the frame width is 176 and frame height is 144 . The total bit rate is 9123kbps.

**Fig 7 : Input video**

**Deep learning layer:**

The deep learning layer will take the sampled input image and it will give us the values of echo, time take to execute, performance, gradient value and validation checks and there by we will get the decision. In this proposed system we will get 407 iterations and time taken by the process is around 6seconds and the performance is 0.240 and the

gradient of the process is 7.85 $e^{-11}$ and the validation checks 0 out of 6. If we compare the values with the previous algorithm the values we got is low and this is the drawback of the previous algorithm and when coming to time the previous algorithm is taking more time to over come this the present system is proposed.

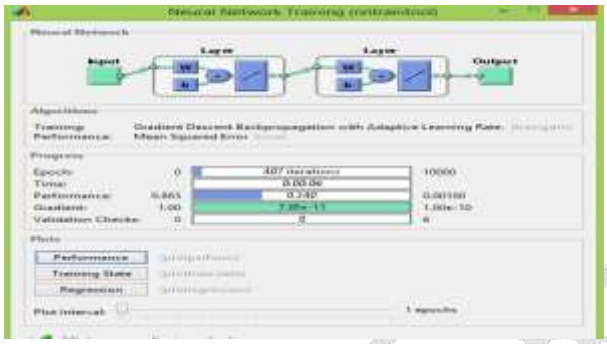

**Fig 8 : Deep learning layer**

**Decision:**

Here we will get the output in the form of decision box where we will see whether the person is authorized are not is shown here.

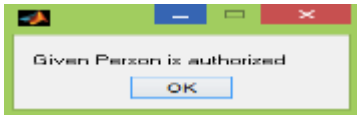


**Fig 9 : Decision**

## CONCLUSION AND FUTURESCOPE

Verifying identities in videos has several applications in social media, surveillance, and law enforcement. Existing approaches have achieved high verification accuracies at equal error rate; however, achieving high performance at low false accept rate is still an arduous research challenge. In this research, a novel video face verification algorithm is proposed which utilizes frame selection and deep learning based feature representation. The proposed algorithm starts with adaptively selecting feature-rich frames from input videos using wavelet decomposition and entropy. The proposed deep learning architecture which combines SDAE joint representation with DBM is used to extract features from the selected frames. The extracted representations from two videos are matched using a feed forward neural network. The results are demonstrated on the challenging Point and Shoot Challenge and YouTube Faces databases. The comparison with state-of-the-art results on both the databases show that the proposed algorithm provides the best results on both the databases at low false accept rate, even with limited training data. Apart from the benchmark protocols of both the databases, several additional experiments have been performed to show the effectiveness of the proposed contributions: (i) joint feature learning in an autoencoder, (ii) sparse and low rank regularization in DBM, and (iii) combination of SDAE and DBM in the proposed architecture.

As a future research, we plan to extend the algorithm for “face recognition in crowd” with multiple subjects in each video. We can also plan for SVM technique for better results in the face verification, we plan to directly train a Siamese network using all available positive and negative pairs from SDAE and training datasets to fully utilize the discriminative information for realizing better performance.

## REFERENCES

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

[2] W. Wang, R. Wang, Z. Huang, S. Shan, and X. Chen, “Discriminantanalysis on riemannian manifold of Gaussian distributions for facerecognition with image sets,” in *Proc. IEEE Conf. Comput. Vis. PatternRecognit.*, Jun. 2015, pp. 2048–2057.

[3] N. M. Khan, X. Nan, A. Quddus, E. Rosales, and L. Guan, “On video based face recognition through adaptive sparse dictionary,” in *Proc.IEEE Int. Conf. Workshops Autom. Face Gesture Recognit.*, May 2015,pp. 1–6.

[4] H. Li and G. Hua, “Hierarchical-PEP model for real-world face recognition, ”in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015,pp. 4055–4064.

[5] H. Li, G. Hua, X. Shen, Z. Lin, and J. Brandt, “Eigen-PEP for video face recognition,” in *Proc. Asian Conf. Comput. Vis.*, 2014, pp. 17–33.

[6] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, “DeepFace: Closingthe gap to human-level performance in face verification,” in *Proc.*

*IEEEConf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1701–1708.

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

[8] 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.

[9] 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.

[10] 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.

[11] 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. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 3554–3561.

[12] 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.

[13] 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.

[14] ] 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.

[15 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.

[16]W. Chengbo, L. Yongping, Z. Hongzhou, and W. Lin, "Classifier discriminant analysis for face verification based on FAR-score normalization," in *Foundations of Computational Intelligence, 2007. FOCI 2007. IEEE Symposium on*, 2007, pp. 476-480.

AUTHORS

Miss. Pallekonda .Pavani completed B.Tech in ECE Department from Intell engineering college, Anantapur. Pursuing Masters in Digital Electronics and Communication System in P.V.K.K engineering college, anantapur.

Mrs. M.Revathi completed B.Tech in ECE Department from Intell Engineering college, Anantapur. Completed Masters in Embedded Systems in Shri Sai Institute of Engineering and Technology Anantapur. Currently working as Assistant Professor in Dept of ECE ,PVKK Institute of Technology ,Anantapur. Having Experience of 3.8 years.

Mr. S.RAVI KUMAR completed B.Tech in ECE Department from G PULLA REDDY Engineering College, Kurnool. Completed Masters in Digital Systems and Computer Electronics in BITS Engineering College, Warangal. Currently working as Associate Professor in Dept of ECE ,PVKK Institute of Technology ,Anantapur. Mail id :ravik.s4u2020@gmail.com