International Journal of Pure and Applied Mathematics
Volume 117 No. 21 2017, 371-389
ISSN: 1311-8080 (printed version); ISSN: 1314-3395 (on-line version)
url: http://www.ijpam.eu
Special Issue

A Deep Learning Approach for Age Invariant Face Recognition

[1]DivyanshuSinha, [2]J.P. Pandey and [3]BhaveshChauhan

[1]Department of Computer Science Engineering,

Uttarakhand Technical University,

Dehradun, India.

sinhadivyanshu63@gmail.com

[2]Department of Electrical Engineering,

KNIT, Sultanpur, India.

[3]Department of Electrical Engineering,

ABESIT, Ghaziabad, India.

Abstract

Soft computing, is as a collection of methodologies, and an important element for constructing a new generation of computationally intelligent systems.This has helped it to achieve great success in solving practical computing problems including Face Recognition. On the other hand, deep learning has become one of the most promising techniques in artificial intelligence in the past decade. The contemporary collection of Soft Computing methodologies, combined with Deep Learning technology reveals a promising direction for complex problem solving.Particularly when applied to face recognition,it improves the overall efficiencyof the algorithm and contributes significantly to the accuracy of recognized faces. Efficient face recognition still encounters several serious challenges despite the fact that there have been numerous recent advances in this field. In this paper we present a system, that utilizes the power of Deep learning through Convolutional Neural Networks, Recurrent Neural Networks and LSTM that enable face recognition, to be easily implemented . Our method uses a deep convolutional network trained to directly optimize the embedding of identity itself, rather than an intermediate bottleneck layer as in previous deep learning approaches. The benefit of our approach is much greater representational efficiency: we achieve state-of-the-art face recognition performance using only 256-bytes per face. On the widely used Cross Age Celebrity Dataset (CACD) dataset, our system achieves a very high degree of accuracy.

Key Words: Deep learning, CNN, RNN, face recognition, aging, LSTM, PCA,LBP.

1. Introduction

Deep artificial neural networks often have far more trainable model parameters than the number ofsamples they are trained on. Nonetheless, some of these models exhibit remarkably small generalizationerror, i.e., difference between "training error" and "test error". At the same time, it iscertainly easy to come up with natural model architectures that generalize poorly.CNN methods have drawn considerable attention in thefield of face recognition in recent years. In this section,we briefly review these CNNs.The researchers in Facebook AI group trained an 8-layerCNN named DeepFace [1]. It is claimed in [2] that one pooling layeris a good balance between local transformation robustnessand preserving texture details. DeepFace is trained on alarge face database which contains four million facial imagesof 4,000 subjects. Another contribution of [1] is the3D face alignment. Traditionally, face images are alignedusing 2D similarity transformation before they are fed intoCNNs. However, this 2D alignment cannot handle out-ofplanerotations. To overcome this limitation, [1] proposesa 3D alignment method using an affine camera model.In [3], a CNN-based face representation, referred to asDeep hidden Identityfeature (DeepID), is proposed. UnlikeDeepFace whose features are learned by one single bigCNN, DeepID is learned by training an ensemble of smallCNN is, used for network fusion.

Soft computing techniquesare based on the human type information processing whichinvolves both logical and intuitiveinformation processing.Conventional computer systems are good for the former,but their capability for the later is far behind that of humanbeings [6].Y. Sun et al.[13]presented a rigorous empirical evaluation of CNNbased face recognition systems. Specifically, we quantitatively evaluate the impact of different architectures and implementation choices of CNNs on face recognition performances on common ground. We have shown that network fusion can significantly improve the face recognition performance because different networks capture the information from different regions and scales to form a powerful face representation .Florian et al.experimented with a similarity transform alignment and noticed that this can actually improve performance slightly. Itis not clear if it is worth the extra complexity[15].Jian et al. revealed deep learning based soft computing in the past decade has been presented. On one hand, the training of deep neural network can be improved with soft computing methods such as genetic algorithm and fuzzy logic[16].

2. Research with Deep Learning Concept

Deep Learning:. Deep learning is the class of machine learning algorithm which has following features:

- It has many cascaded layers for feature extraction. The output of one layer serves as input of second layer.

- Higher level features are derived from lower level features to form hierarchical representation.

The layers used in the deep learning are the hidden layers of the artificial neural network. One more advantage of deep learning is that layers selects best features. Neural networks are trained using the gradient back-propagation method. To explain this it is about updating the weight of a layer as the derivative of previous layer. Layers are the representation of lower to higher levels concepts.

3. Research Methodology

Research Methodology Used (Hybrid CNN)

While training dataset with multilayer CNN (Convolutional Neural Network) challenging problem was to find best parameters to get high accuracy. To overcome this problem adaptive CNN is proposed in this work which leads to higher accuracy than CNN. Details of proposed approach are depicted below:

Work weighted approach is used in which fine tuning of weights of CNN is achieved. Word $\hat{a}$ with the highest number of words is selected as final recognition.

$$\hat{a} = arg \max_{w \in L} \sum_{i=1}^{N} \alpha_i \, . P_i(a)$$

α_i is the weight assigned to C_i and $P_i(a)$ is an indicator function which takes the value 1 if the argument is equal to word recognized by C_i otherwise 0. Following steps are implemented to achieve our objective.

Step 1: Break the image into overlapping image tiles.

Step 2: Feed each image tile into a small neural network

Step 3: Save the results from each tile into a new array

Step 4: Perform Down sampling

Step 5: Make a prediction

Three different layers of CNN are studied and used. These layers are:

1. Input layer which is the first layer of CNN and take input from preprocessing unit. Following is the block diagram of input layer.

Input layer take the input from last unit in the form of matrix.

In the proposed algorithm total parameters are 5,733,831where trained parameters are 5,733,767and non- trainable parameters are 64.

A. Convolutional Layers

Generally convolutional can be represented by the following equation as

$$(f * g)(t) \stackrel{\text{def}}{=} \oint_{-\infty}^{\infty} f(T)\, g(t-T) dT \quad (1)$$

Let layer l be a convolutional layer. Suppose that we have some M × M square neuron nodes which is followed by our convolutional layer. If we use an N × N filter W then convolutional layer output will be of size (M −N +1) × (M −N +1) which produce k-feature maps that illustrated in Fig. 1. Convolutional layer acts as a feature extractor that extracts salient features of the inputs such as corners, edges, endpoints. In or- der to compute the pre-nonlinearity input to some unit. Then, the input of layer l − 1 comprises is computed as:

$$Y_i^l = B_i^{(l)} + \sum_{a=1}^{M} \sum_{b=1}^{M} W_i\, X_{(i+a)(j+b)}^{l-1} \qquad (2)$$

Where $B_i^{(l)}$ is a bias matrix and $W_i^{(l)}$ is the filter of size N × N. Then, the convolutional layer applies its activation function as:

$$Z_i^l = \sigma(B_i^l) \qquad (3)$$

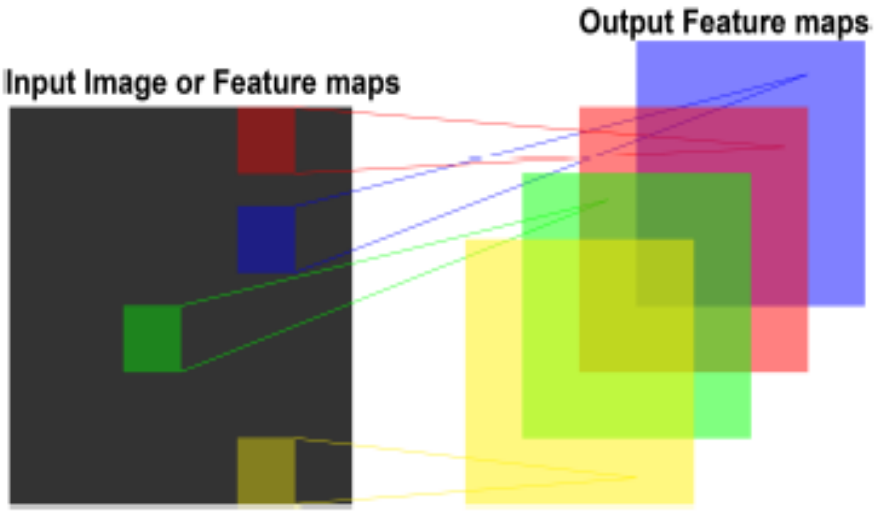


Figure 1: Illustration of a Single Convolutional Layer that Produces k-Feature Maps

B. Activation Function

Activation functions are there to give deep nets the powers they are advertised to have. Without activation functions, deep nets lose a bulk of their representation learning power or just miserably fail to deliver any computations at all.

Mathematically it is described as max(0,x). Softplus is a smoothed version of

ReLu whose equation for optimization is written as

$$df(x) = \frac{d(\ln(1+e^x))}{dx} = \frac{e^x}{1+e^x} \quad (4)$$

We apply the ReLu nonlinearity as an activation function to the output of every convolutional layer and fully connected layer. ReLu is more biologically plausible it can be engaged as better activation functions than the widely used logistic sigmoid and hyperbolic tangent functions. The ReLu increases the nonlinear properties of the decision function and of the overall network without affecting the receptive fields of the convolution layer. The use of Re- LUs map more plausibly to biological neurons makes the training of deep neural network significantly faster and improves its generalization ability.

After applying activation function we again apply Convolutional layer and get a matric of 32, 3x3.

C. Batch Normalization

Training Deep Neural Networks is complicated by the fact that the distribution of each layer's inputs changes during training, as the parameters of the previous layers change. This slows down the training by requiring lower learning rates and careful parameter initialization, and makes it notoriously hard to train models with saturating nonlinearities. We refer to this phenomenon as internal covariate shift, and address the problem by normalizing layer inputs. Our method draws its strength from making normalization a part of the model architecture and performing the normalization for each training mini-batch. Batch Normalization allows us to use much higher learning rates and be less careful about initialization. It also acts as a regularizer, in some cases eliminating the need for Dropout. Applied to a state-of-the-art image classification model, Batch Normalization achieves the same accuracy with 14 times fewer training steps, and beats the original model by a significant margin.

$$\sigma_\beta^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_\beta)^2 \quad (5)$$

D. Max Pooling Layers

- CNN layers are modified using back propagation algorithm and in the forward path explicit computation with all weights are performed.
- In the forward pass, perform explicit computation with all weights,$w_1, w_2 \ldots . w_9$(some of them are same).
- In the backward pass, compute the average of gradient $\frac{\partial J}{\partial w_1}$,….., $\frac{\partial J}{\partial w_9}$, for all the weights.
- When updating the weights, use the average of the gradients from the shared weights, e.g$w_1 = w_1 - \alpha\left(\frac{\partial J}{\partial w_1} + \frac{\partial J}{\partial w_4} + \frac{\partial J}{\partial w_7}\right)$, $w_4 = w_4 - \alpha\left(\frac{\partial J}{\partial w_1} + \frac{\partial J}{\partial w_4} + \frac{\partial J}{\partial w_7}\right)$ and $w_7 = w_7 - \alpha\left(\frac{\partial J}{\partial w_1} + \frac{\partial J}{\partial w_4} + \frac{\partial J}{\partial w_7}\right)$

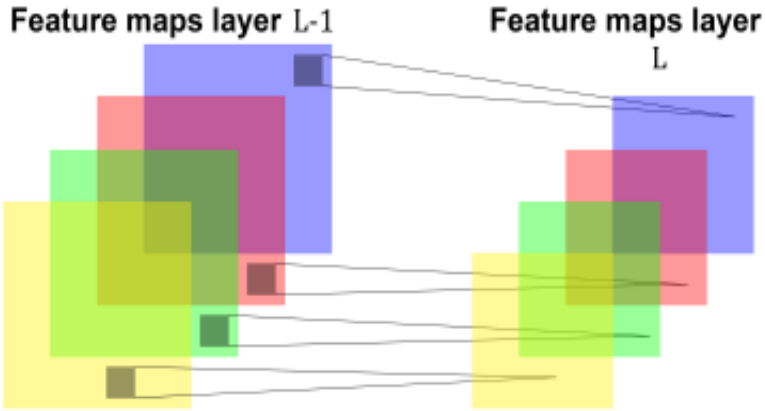


Figure 2: Illustration of Pooling Layer that Produces the Same k-Feature Maps

E. Dropout Layer

A dropout layer is used for regularization where you randomly set some of the dimensions of your input vector to be zero with probability keep_probkeep_prob. A dropout layer does not have any trainable parameters i.e. nothing gets updated during backward pass of backpropagation. To ensure that expected sum of vectors fed to this layer remains the same if no dropout was applied, the remaining dimensions which are not set to zero are scaled by 1/keep_prob.

$$w_{t+1} = w_t - \eta_t \nabla l(w_t{}^T(x_t \; o \in_t), y_t) \quad (6)$$

4. Proposed Work with Deep Learning Experimental Justification

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	(None, 250, 249, 1)	0	
conv2d_1 (Conv2D)	(None, 250, 249, 256)	2560	input_1[0][0]
activation_1 (Activation)	(None, 250, 249, 256)	0	conv2d_1[0][0]
conv2d_2 (Conv2D)	(None, 248, 247, 32)	73760	activation_1[0][0]

batch_normalization_1 BatchNorm (None, 248, 247, 32) 128
conv2d_2[0][0]

activation_2 (Activation) (None, 248, 247, 32) 0 batch_normalization_1[0][0]

max_pooling2d_1 (MaxPooling2D) (None, 124, 123, 32) 0
activation_2[0][0]

dropout_1 (Dropout) (None, 124, 123, 32) 0 max_pooling2d_1[0][0]

reshape_1 (Reshape) (None, 124, 3936) 0 dropout_1[0][0]

dense_1 (Dense) (None, 124, 937) 3688969 reshape_1[0][0]

gru1 (GRU) (None, 124, 128) 409344 dense_1[0][0]

gru1_b (GRU) (None, 124, 128) 409344 dense_1[0][0]

merge_1 (Merge) (None, 124, 128) 0 gru1[0][0]
gru1_b[0][0]

gru2 (GRU) (None, 124, 128) 98688 merge_1[0][0]

gru2_b (GRU) (None, 124, 128) 98688 merge_1[0][0]

merge_2 (Merge) (None, 124, 256) 0 gru2[0][0]
gru2_b[0][0]

flatten_1 (Flatten) (None, 31744) 0 merge_2[0][0]

c1 (Dense) (None, 30) 952350 flatten_1[0][0]

Total params: 5,733,831
Trainable params: 5,733,767
Non-trainable params: 64
Output obtained from the above is passed to Flatten and dense layer and at the end you can see the total number of parameters equals to 5,733,831from which5,733,767are trainable and 64 are non- trainable.

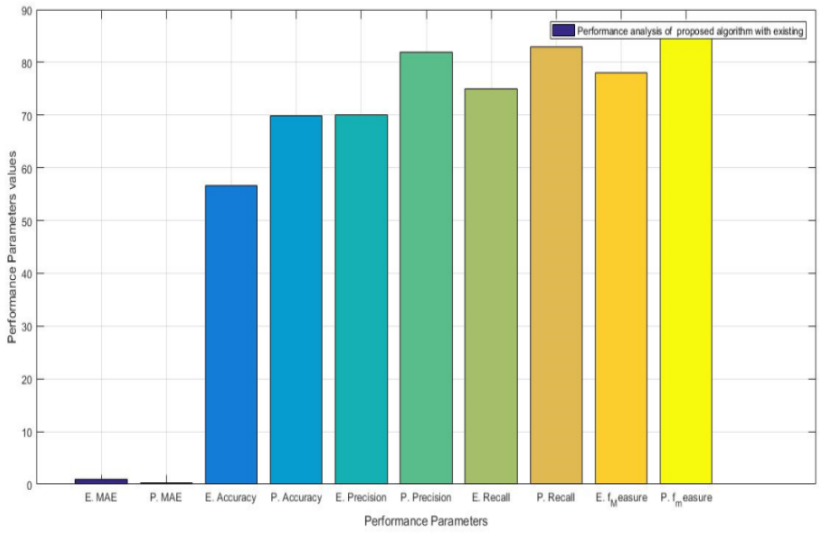


Figure 3: Performance Parameter of the Research

We have used LBP for feature extraction& Feature space representation of LBPH is done using PCA which has also been used for Dimensionality Reduction. A face image is first divided into small regions where Local Binary Pattern (LBP) histograms are extracted andthen concatenated into a single feature vector. This feature vector will further reduce the dimensionality scope by using the well establishedPrinciple Component Analysis (PCA) technique.

The general idea behind this approach is to extract the main information in the training set as represented by some template images that capture most of the variability in the data. This is done by calculating the vectors which is the eigenvectors of the covariance matrix for the training set, that best represent this small region of image space. By using only the eigenvectors with eigenvalues that is above some threshold defined a new data set is obtained with less dimensions, but still with the most important characteristics of the original data. So the main idea of PCA, used for face recognition, is that the original data is transformed to a different space, with fewer dimensions, in which it is easier to measure the relevant differences and similarities between them.

LBP can easily be combined with PCA to reduce the length of the feature vector, which leads to improvement in recognition performance. LBP or local binary pattern is a visual descriptor used for classification. Basically it is a powerful feature for texture classification. The concept of basic LBP operator is

to compare the center pixel value with its eight neighbors in a fixed 3x3 pixel window. If the neighborhood pixel valuesare greater than or equal to center value it is assigned alabel 1 otherwise 0.

Following are the steps for calculating features:

- If the image is of RGB or true color, convert it into grayscale using function rgb2gray.
- The parameters of LBP are number of sample points in a region and radius of the region.
- Size of LBP label is calculated using equation $\mathrm{L} = 2 * \mathrm{R} + 1$, where L is he size of LBP label.
- Using the input image and size of LBP label row_max and column_max is calculated and iteration is run using these constants and label size and LBP images are formed.

Mathematical Equation

Let T be a variable and $p_1, p_2, p_3$are local neighborhood of pixel.

$T = t(g_c, g_0, p_{p-1})$where gc is the gray value of a central pixe l and r est are the gray value of neighbouring pixel.

LBP equation follows the neighboring mathematical equation

Let P is the neighbouring gray value. R is the radius of concerned circle.

g_p: gray level of neighbourhood pixel

g_c:gray level of central pixel

$$LBP_{P,R}(i,j) = \sum_{P=0}^{P=P-1} S(g_p - g_c)^{2^P}$$

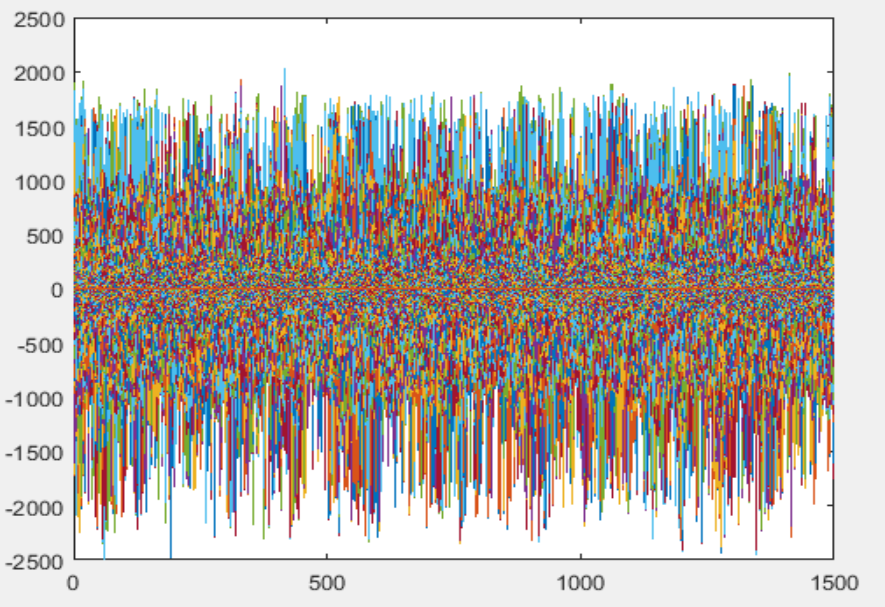


Figure 4: Features Extracted & Dimensionality Reduction using PCA & LBP

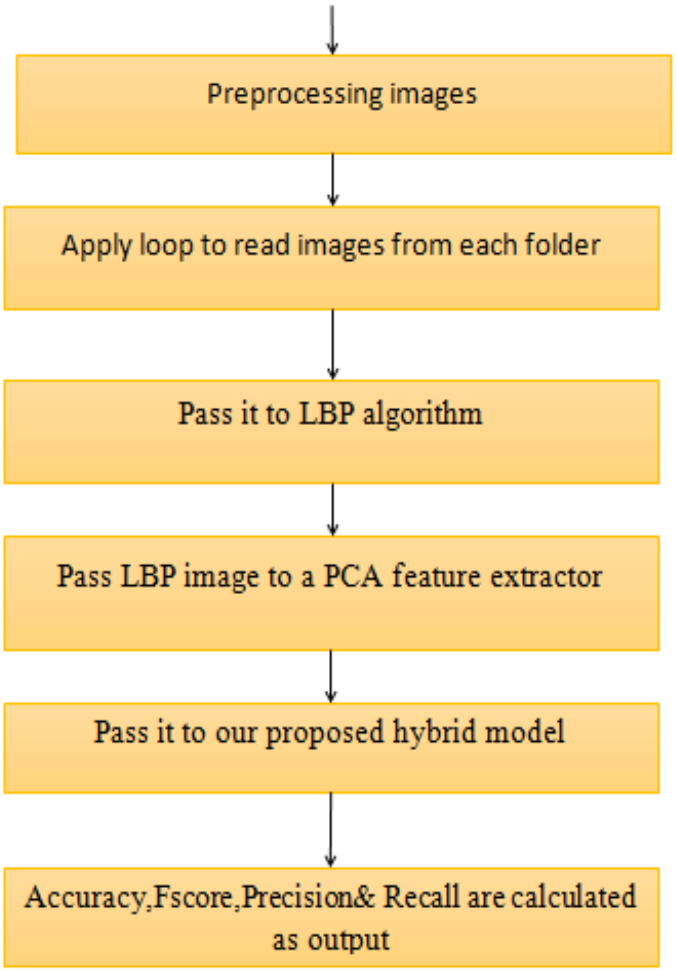


Figure 5: Flow Chart of the Proposed System

Convolutional neural network:- A feed forward network is defined as a network having no cycles contained with it. When training a feed forward network information is passed into the net and resulting classification is compared to the known training sample. In simple words CNN is a feed forward network and is trained using back propagation method.

LSTM:-It is a special type of RNN is a neural network having loops. This loop allows the information from the previous passes which acts as a memory.

Most convolution Layer

Shape of input = (250 250)
Numbers of convolution filters = 256
OIP shape = [250 250 **256**]
Size of each filter = 3*3=9
Parameters obtained 256*9+256 = 2560 fit the products with better do for good lifting

Stochastic gradient descent

$\theta_{t+1}=\theta_t - \alpha\delta l(\theta 1)$

θ- weights changed according to the gradient of the loss with respect to each (θ).

α- learning rate.

Further to improve zig,zagmomentum as applied. And introduced a parameter μ

$V_{t+1} = \mu v_t - \alpha\delta L(\theta 1)$

$\theta_{t+1}= \theta_t - v_{t+1}$

Substitution value of v_{t+1}

$\theta_{t+1}= \theta_t - [\mu vt - \alpha\delta L(\theta 1)]$

Convolutional Layer

ADAM:-It is based on first order moment and second order moment. Let us supposefirst ordermoment is given by it and second order by it.

Moment in simple words,is thequality measure. Generally firstmoment is mean andsecond moment around mean isvariance .

Equations of ADAM

$$m_t+1 = y_1 m_t + (1-y_1)\nabla l \ (\theta_1)$$
$$g_t+1 = y_2 g_t + (1-y_2)\nabla l \ (\theta_2)$$

ADAM delta is exponential decaying average of gt.

which issecond moment ofgradient

$G_{t+1}\gamma_{gt} + (1-\gamma)\ \nabla l\ (\theta)^2$

Image Details

Number of classes= 30

Rows of each image= 250

Columns of each images= 250

Loading mat file in Python

- importscipy . co as sio
- f=sio. Load mat (name of file)
- data = f (all features mat) # this is structured from of data
- x train = np Array (data) # training part. Training data
- X _ test = np array (data)
- Y train and Y Test is defined

Reshaping in python

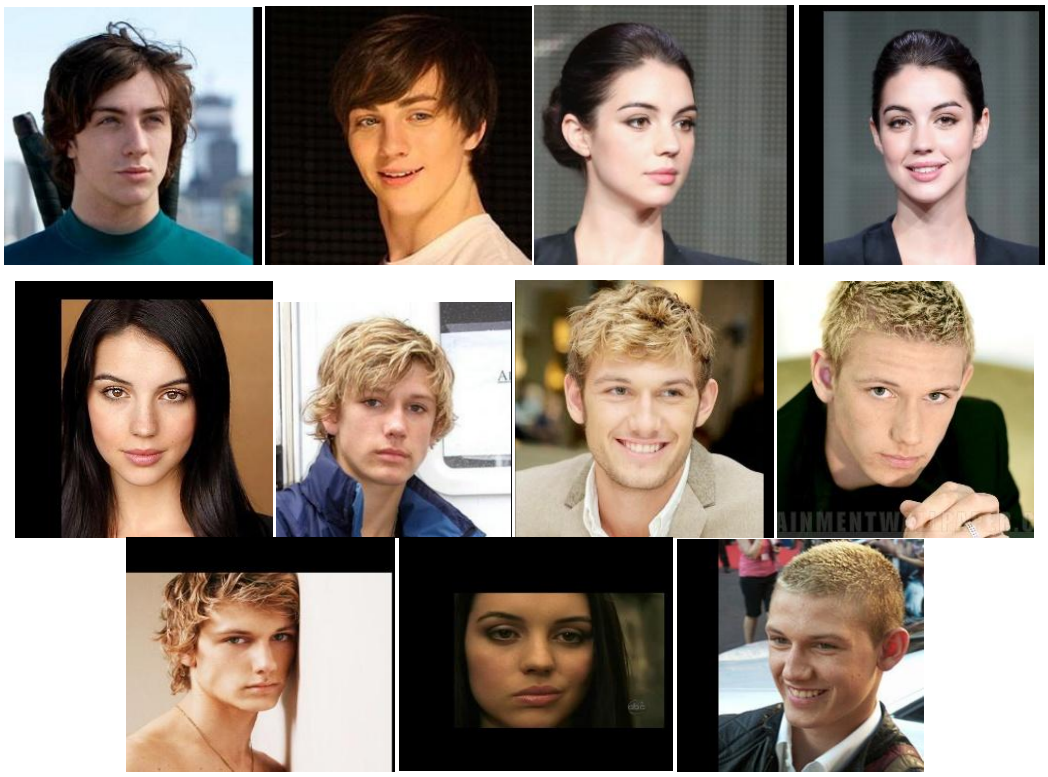
Shape of x-train = 1500x 62260

But we need shape as = [1500x250x249x1]

To archive this we need tochange shape which is done by shape f (x)

Input layer: Inputlayeris an image having pixels equal to **250*250 = 62250.**

5. Results Explanation& Discussions



Face Images from CACD

Cross-Age Celebrity Dataset (CACD) contains 163,446 images from 2,000 celebrities collected from the Internet. The images are collected from search engines using celebrity name and year (2004-2013) as keywords.

Explanation

Loss: Loss is not in percentage as opposed to accuracy but it is summation of error made for each in training. If during the epochs the loss decreases, such model is defined as fit model.

Accuracy: It is defined in the sense how the model parameters are learned. Basically accuracy means how accurately prediction value matched with the real value and it is inversely proportional to the loss rate.

Recall: This is very important parameter and is calculated after creating the confusion matrix and is used to analyse the percentage of selected cases from correctly classified cases. High value refers to robustness of the work.

Precision: This is again a parameter related to recall but instead of correctly classified cases, it refers to overall selected cases and out of it how many are classified correctly. Overall Precision refers to accuracy and is also interrelated to recall.

F_measure: This is another parameter which is a hybrid of both Recall and F_measure.

To further evaluate our algorithm we use Precision, Recall &Fscore

For our algorithm these values are:

Table 1: Parametric Values

Parameters	Values
Precision	0.99934640522875817
Recall	0.9993333333333333
f-score	0.99933326665999933

Table 2

Epoch /10	ETA	Loss	Accuracy	Val_loss	Val_acc
1	29	4.7646	0.0353	4.4712	0.098
2	26	3.5888	0.0973	2.6816	0.2707
3	26	2.8732	0.2267	1.9592	0.5113
4	26	2.0983	0.4193	1.1784	0.7427
5	26	1.4928	0.6067	0.8351	0.878
6	26	0.9525	0.7913	0.4331	0.9813
7	26	0.6243	0.9047	0.2715	0.9987
8	26	0.41	0.9567	0.1937	0.9993
9	26	0.2583	0.9893	0.1278	0.9993
10	26	0.1939	0.99	0.087	0.9993

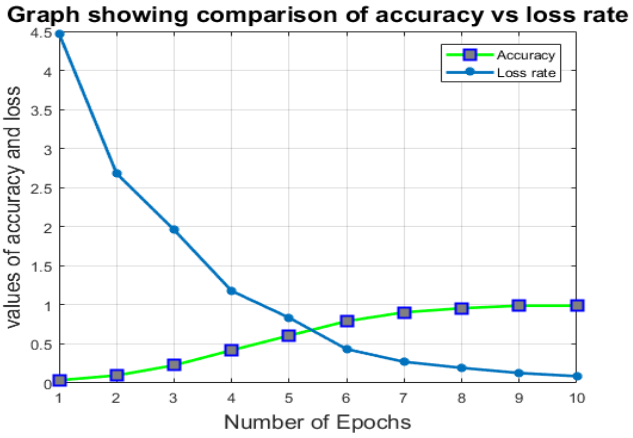


Figure 6: Comparison Graph of Accuracy vs Loss Rate

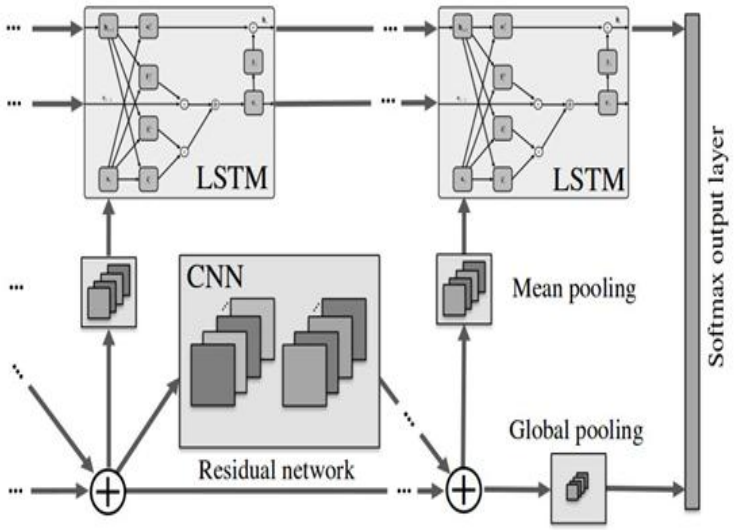


Figure 7: Our Hybrid Model Comprising of CNN & LSTM

6. Discussions

In this work, the combination of CNN & LSTM depicts that with the increase in the number of epochs neuron starts learning about the data in terms of updated weights and their loss starts decreasing with every epoch. Error is minimized using the concept of back propagation method and by the time we reache 10th epoch accuracy has reached 99.93% which could also been seen in the value of other performance parameters. To supplement our algorithm we calculate confusion matrix since this is a multi label class so confusion matrix is calculated using the argument of both testing data and prediction data.

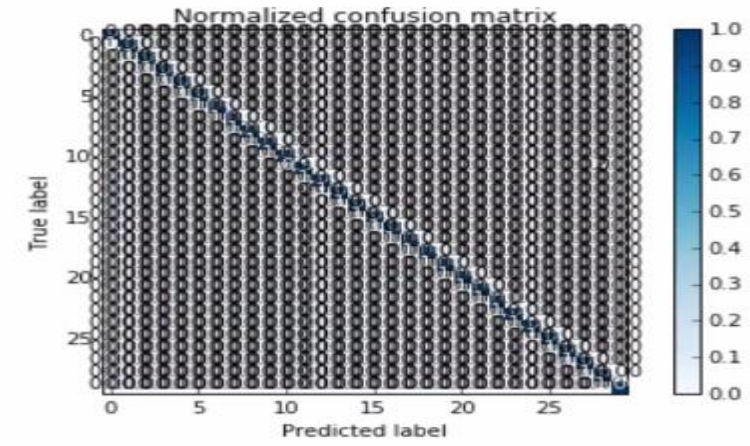


Figure 8: Confusion Matrix of True Label & Predicted Labelhis Information is Pictorially Depicted Through the Graph which Shows a Comparison of Values of Accuracy & Loss Vs Epochs

In our experiments, dropout did significantly improve the performance of our CNNs. Softmax is used in the last layer for predicting one of K (the number of subjects in the context of face recognition) mutually exclusive classes.How to design the architecture of a neural network is an open problem.

Though the high level structure of CNN usually starts with a number of convolutional layers, followedby a (fewer) number of fully-connected layers, the choices of filter size, number of neurons per layer etcare usually determined by trial-and-error.

Instead of reporting the final design directly, we show some inferior designs on the way to finding the optimal one. Our strategy is as follows: we start from a relatively small network, then extend it (by adding more neurons and/or layers) while the performance improves, and stop when it gets worse or becomes constant as the CNN size increases, which indicates that overfitting occurs.

Comparison with State of the Art

The proposed work is compared with the other state of art method. The difference among them is that I have introduced the concept of bright preserving fuzziness to the images before training them as compared to the previous work. This reduces mean square error and improves peak signal to noise ratio.

We finally compare our full method with non-commercial state of the art methods.

The results in Fig 8 show that our method is better than [9, 10, 11, 12, 13]. However, [11] needs to detect many accurate facial landmarks to assist feature extraction, we do not; Compared with the fisher vector face [12], the feature dimensionality of our model is much lower. [13] generates a large number of new pairs to train the model, just as we do.

Table 3: Comparison with State-of-the-Art Methods Under 'Unrestricted, Label-Free Outside Data'method Accuracy

Method	Accuracy
LBP multishot [9]	0.8517 ± 0.0061
LDML-MkNN [10]	0.8750 ± 0.0040
High-dim LBP [11]	0.9318 ± 0.0107
Fisher vector faces [12]	0.9303 ± 0.0105
ConvNet+RBM [13]	0.9175 ± 0.0048
Fuzzy SVM Based Hybrid Approach [14]	0.85
Our Proposed approach	0.9993

In the proposed method we achieve a classification accuracy of 99.93% on the CACD dataset.

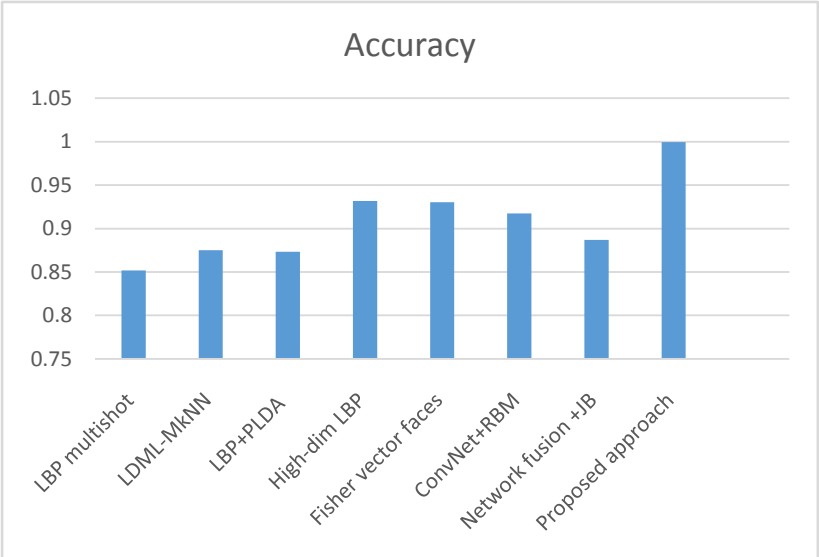


Figure 9: Comparison with State of the Art Techniques

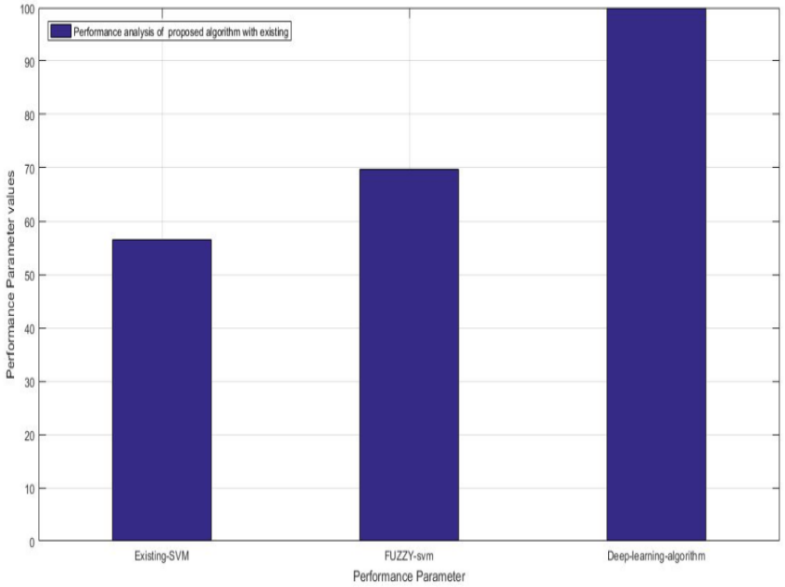


Figure 10: Performance Evaluation of The work with Existing Research

7. Conclusion

Here we have presented a fusion of Soft Computing & Deep Learning to rigorously evaluate our approach in Face Recognition. Combination of Convolutional Neural Networks, Recurrent Neural Networks & LSTM significantly improves the performance of this algorithm in comparison to other approaches as indicated above. This makes it standout from other methods which require additional post-processing such as concatenation of multiple models and PCA, as well as SVM classification. Our end-to-end training both

simplifies the setup and shows that directly optimizing a loss relevant to the task at hand improves performance.

Future work will focus on better understanding of the error cases, further improving the model, and also reducing model size and reducing CPU requirements. We will also look into ways of improving the currently extremely long training times, e.g. variations of our curriculum learning with smaller batch sizes and offline as well as online positive and negative mining.

References

[1] Taigman Y., Yang M., Ranzato M., Wolf L., Deepface: Closing the gap to human-level performance in face verification, IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2014), 1701–1708.

[2] Taigman Y., Wolf L., Hassner T., Multiple one-shots for utilizing class label information, In BMVC, (2009).

[3] Sun Y., Wang X., Tang X., Deep learning face representation from predicting 10,000 classes, IEEE Conference on Computer Visionand Pattern Recognition (CVPR) (2014), 1891–1898.

[4] Tanvi N.G., Kaur M., A review of soft computing techniques in biometrics, In International Conference on Recent Advances in Engineering & Computational Sciences (2015), 1–4.

[5] Chattopadhyay S., Soft computing techniques in combating the complexity of the atmosphere- a review, (2006).

[6] Ho W., S G.T.. Ho P.J., Lau H.C.W., A hybrid geneticalgorithm for the multi-depot vehicle routing problem, Engineering Applications of Artificial Intelligence 21 (4) (2008), 548–557.

[7] Fan Y., Guo Z., Yin Y., Fan Y., Yin Y., Semg-based neuro-fuzzy controller for a parallel ankle exoskeleton withproprioception, International Journal of Robotics & Automation 26 (4) (2011), 450–460.

[8] Aldobhani A.M.S., John R., Maximum power point tracking of pv system using anfis prediction and fuzzy logic tracking, Lecture Notes in Engineering & Computer Science 2169 (1) (2008), 113–122.

[9] Taigman Y., Wolf L., Hassner T., Multiple one-shots for utilizing class label information, In BMVC, (2009).

[10] Guillaumin M., Verbeek J., Schmid C., Is that you? Metric learning approaches for face identification, IEEE 12th International Conference on Computer Vision (2009), 498–505.

[11] Chen D., Cao X., Wen F., Sun J., Blessing of dimensionality: High-dimensional feature and its efficient compression for face verification, IEEE Conference on Computer Vision and Pattern Recognition(CVPR) (2013), 3025–3032.

[12] Simonyan K., Parkhi O.M., Vedaldi A., Zisserman A., Fisher Vector Faces in the Wild.In Conference on British Machine Vision, (2013).

[13] Sun Y., Wang X., Tang X., Hybrid deep learning for face verification. IEEE International Conference on Computer Vision (ICCV) (2013), 1489–1496.

[14] Sinha D., Pandey J.P., Chauhan B., Fuzzy SVM Based Hybrid Approach for Age Invariant Face Recognition Journal of Advanced Research in Dynamical & Control Systems, (2017).

[15] Florian Schroff, Dmitry Kalenichenko, JamesPhilbin, FaceNet: A Unified Embedding for Face Recognition and Clustering, (2015).

[16] Jian Zhang, Chongyuan Tao, A Review of Soft Computing Based on Deep Learning, International Conference on Industrial Informatics - Computing Technology, Intelligent Technology, Industrial Information Integration, (2016).