



HOG Based Facial Recognition Approach Using Viola Jones Algorithm and Extreme Learning Machine

Khushwant Sehra¹, Ankit Rajpal²(✉), Anurag Mishra²,
and Girija Chetty³

¹ University School of Information, Communication and Technology,
Guru Gobind Singh Indraprastha University, New Delhi 110078, India

² Deen Dayal Upadhyaya College, University of Delhi,
New Delhi 110078, India
ankit30sep@gmail.com

³ Faculty of Science and Technology, University of Canberra,
Bruce, ACT 1617, Australia

Abstract. Extreme Learning Machine has attracted widespread attention for its exemplary performance in solving regression and classification problems. It is a type of single layer feed-forward neural machine which relies on randomly allocating the input weights and hidden layer biases. Through this, the ELM has been found to possess running time spans which are within millisecond regime. It does not require complex controlling parameters which makes its implementation elementary. This paper investigates the performance of employing Extreme Learning Machine as a classifier to be used for the face recognition problem. Viola Jones algorithm has been employed to detect and extract the faces from the dataset. Finally, Histogram of Oriented Gradients (HOG) features are extracted which form the basis of classification. The scheme so presented has been tested on standard face recognition datasets from AT&T and YALE. The resulting training/testing time spans of the whole scheme range from milliseconds to seconds, dictating the compatibility of ELM with real-time events.

1 Introduction

The advent of facial recognition in the field of pattern recognition has found a great range of applicability especially for the purpose of cyber investigations. This has been possible due to the progressions made in the analysis and modelling techniques. Increased demand for secured systems has led the researchers to find solutions in terms of accessing control, verifying identities, securing cyber physical systems, internet communications, computer entertainment and establishing surveillance systems that are sturdy and impenetrable [1–3].

Alongside automatic facial recognition systems, automatic processing of digitalized content (like videos and images) has also been achieved due to the low-cost computing systems. As manipulations with identity cards and encroachments into the virtual/physical areas were creating a nuance, it was realized that there was a dire need to have systems that were reliable and could recognize individuals accurately. A number of

advancements were made including biometric authentication, computer-human interactions, machine learning, surveillance etc., thereby leading to a natural discourse of research and development in the field of automatic face recognition.

Identification processes of biometric authentication like iris recognition are highly advanced and individualistic but are intrusive in nature. With the evolution of digital technologies and the challenges offered by human identification and surveillance systems, research in the area of face recognition has become imperative as convenient, natural and non-intrusive in nature [4]. In order to discern facial features correctly, various facial recognition systems are available. However, they do not provide the precision needed for a reliable recognition to be made. Thus, better face detection algorithms form a pre-requisite requirement for pattern recognition and computer vision applications.

Further, issues due to variations in the illumination have greatly reduced the potential of facial recognition systems. This is because there is a marked difference in the facial images of the same individual obtained under different illumination. The existing systems are highly sensitive to light variations [5]. Facial images taken under a condition where illumination was un-controlled suffered a non-uniform illumination. To cope with this, certain adaptive techniques are being used. These normalization techniques emend the illumination and restore the features of an image to its original form. Examples of illumination normalization techniques are: Logarithmic Transform (LT), Gamma correction (GC) and Histogram equalization etc. [5, 6]. Extreme Learning Machine (ELM) and its kernelized variant K - ELM has been previously employed for facial recognition schemes by Zong et al. [7, 8]. An attempt to enhance the facial classification using ELM based on facial views has been demonstrated by Iosifidis et al. [9]. Rujirakul et al. in [10] demonstrate the use of histogram equalization coupled with principal component analysis (PCA) in hybrid with ELM for facial recognition. Independent Component Analysis (ICA) has also been used in conjunction with hybrid of Standard Particle Swarm Organization (SPSO) and ELM to demonstrate recognition rates upto 93% [11]. Similar schemes employing Linear Discriminant Analysis (LDA) and multi-class support vector machines have also been reported [12–14]. An effort has also been made in using local difference binary (LDB) descriptors and fuzzy logic with histogram of oriented gradients (HOG) for efficient facial recognition systems [15, 16].

In this paper, Viola-Jones algorithm is applied to identify the regions corresponding to the face of a subject which is ultimately used for extraction of Histogram Oriented Gradients (HOG) features. HOG feature selection is used as a pre-processing technique as it corrects the overall brightness of a face image to a pre-defined canonical form which essentially discards the consequence of varying lighting. For each image in the dataset, HOG extracts crucial features which form the basis for training of the neural machine.

The paper is structured as follows. Section 2 describes the basics of ELM. Section 3 gives an outline of Viola-Jones Algorithm. Section 4 gives a brief introduction of HOG. Section 5 gives an insight into the face recognition approach that has been adopted in this paper. The results have been summarized in Sect. 6 and finally, the paper has been concluded in Sect. 7.

2 Extreme Learning Machine

The Extreme Learning Machine (ELM) is a single layer feed forward neural network [17–19]. Unlike traditional neural machines, application of an ELM is simple. It does not require controlling parameters like learning rate, stopping iterations etc. which are technically complex. It works on the basis of random allocations of input weights and hidden layer biases. This necessitates probability distribution functions that are continuous in nature. Using an inverse method i.e. Moore-Penrose generalized pseudo inverse, output weights are determined [18].

2.1 The ELM Model

Let us consider a set of N training samples (x_i, y_i) where $x_i \in \mathbb{R}^n, y_i \in \mathbb{R}^m$ and $i = 1, 2, \dots, N$. The number of hidden neurons be denoted as \hat{N} .

If ‘ g ’ is the activation function then $g: \mathbb{R} \rightarrow \mathbb{R}$. The output of the system [17] can then be given as:

$$\sum_{k=1}^{\hat{N}} \beta_k g(w_k x_i + b_k) = o_i \forall i \in 1, 2, \dots, N \tag{1}$$

Here w_k is the weighting vector that connects the k^{th} hidden neuron with the input nodes. Similarly, β_k is a weighting vector which connects the k^{th} hidden neuron to the output node. b_k represents the threshold bias of the k^{th} hidden neuron.

As mentioned before, the weighting vectors are chosen randomly as per the continuous probability distribution function. The neural network with \hat{N} hidden neurons and activation function $g: \mathbb{R} \rightarrow \mathbb{R}$ approximates samples with zero error. The Eq. (1) can thus be written as:

$$\sum_{k=1}^{\hat{N}} \beta_k g(w_k x_i + b_k) = y_i \forall i \in 1, 2, \dots, N \tag{2}$$

Thus, we have,

$$H\beta = Y \tag{3}$$

where,

$$H = \begin{bmatrix} g(w_1 x_1 + b_1) & \dots & g(w_{\hat{N}} x_1 + b_{\hat{N}}) \\ \vdots & \dots & \vdots \\ g(w_1 x_N + b_1) & \dots & g(w_{\hat{N}} x_N + b_{\hat{N}}) \end{bmatrix}_{N \times \hat{N}} \tag{4}$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_N^T \end{bmatrix}_{\widehat{NXm}} \quad \text{and} \quad Y = \begin{bmatrix} y_1^T \\ \vdots \\ y_N^T \end{bmatrix}_{NXm} \tag{5}$$

The matrix H represents the hidden layer output matrix. The solution of the above system as given by Huang et al. [13] is:

$$\widehat{\beta} = H^\dagger Y \tag{6}$$

where, H^\dagger is the Moore-Penrose generalized inverse of the hidden-layer output matrix H.

3 Viola Jones Algorithm

Viola Jones algorithm is an object detection framework put forward by Paul Viola and Michael Jones in 2001 [20]. Viola Jones object detector is based on a binary classifier that produces a positive output when the search window consists of the desired object otherwise it returns a negative output. The classifier may be used a number of times as the window slides over the image under test.

The binary classifier used in the algorithm is realized using several layers of hierarchy forming an ensemble classifier [21]. The said classifier operates by classifying images based on value of simple features. This is observed to operate much faster than a system which basis classification on a pixel-based system [20]. The Viola Jones algorithm exercises control over three features as dictated by Viola et al. in [20] viz., Two-Rectangle Feature, Three-Rectangle Feature, and Four-Rectangle Feature. The framework put forward by the group is noted to have following stages: (i) Haar Feature Selection, (ii) Integral Image Generation, (iii) Adaboost Training, and (iv) Cascading Classifiers. This is represented as a flowchart in Fig. 1.

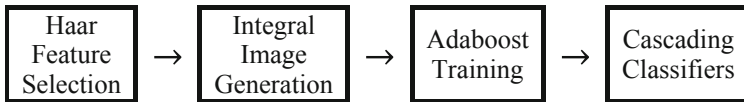


Fig. 1. A depiction of the Viola-Jones Algorithm involving four important stages

The Haar feature selection is computed through Haar basis functions that are based on the three features as listed above and generally include pixel summation of involved adjacent rectangular areas and then calculates the difference between these sums. A depiction of Haar features relative to the corresponding detection window is shown in Fig. 2.

The integral image is then created which is used to evaluate the rectangular features in a constant time. Since the number of features can vary greatly, Adaboost or Adaptive Boosting algorithm is used to select best features and to train the classifiers that use them. This is responsible for creation of a “strong” classifier which is viewed as a linear

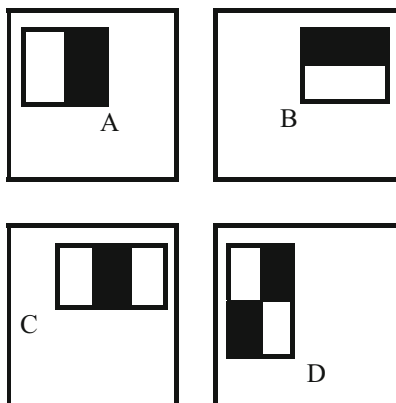


Fig. 2. Depiction of rectangle features shown relative to the detection window.

weighted combination of simple “weak” classifiers. Finally, in cascading, each stage consisting of “strong” classifiers are grouped into several stages. Each stage is responsible for determining whether a sub-window consists of a face or not as depicted in Fig. 3. The algorithm described is implemented using a MATLAB inbuilt routine as described in [22].

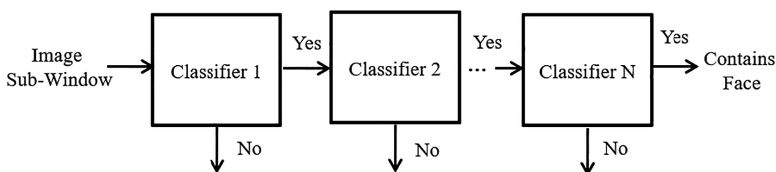


Fig. 3. Depiction of the working flow of classifiers in Viola-Jones algorithm

4 Histograms of Oriented Gradients

Histograms of oriented gradients (HOG) finds applications in object and pattern recognition domain as it is capable of extracting crucial information even from the images that are obtained under garbled environments [23]. It is therefore well suited for tackling the facial recognition problem. The feature extraction process of HOG is based on extracting information about the edges in local regions of a target image [23]. Simply put, HOG feature extraction is primarily the characterization of the orientation and magnitude values of the pixels in an image [24]. That is, it defines an image in terms of groups of local histograms that point to local regions of an image.

The features of HOG can be seen on a grid of rose plots spaced uniformly. The grid dimensions depend upon the size of the cell and image. Thus, every rose plot depicts the gradient orientations distributed in a HOG cell. In a cell histogram, the length of the

petals in a rose plot refers to the contribution of every orientation. For the gradient directions, the plot indicates the directions of the edges that are normal. MATLAB inbuilt routine are applied using HOG feature extraction [25].

Thus, in a portion of image with 9 cells (Fig. 4), the HOG feature extraction routine takes input a block of $(m \times n)$ cells and arranges them in a vector as depicted in Fig. 5.

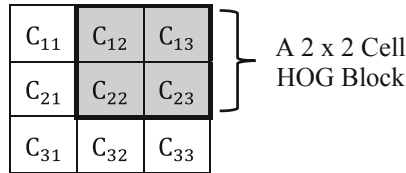


Fig. 4. A portion of image realized using equal sized cells. Each cell consists of pixel values of that portion of the image.

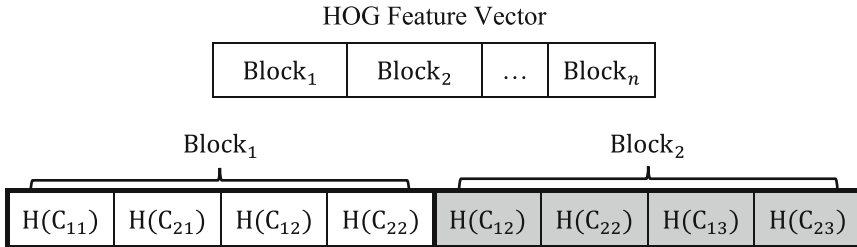


Fig. 5. Depiction of composition of HOG feature vector. $H(C_{ij})$ represents the cell histogram at (i, j) position

5 Algorithmic Description of the Proposed Scheme

The facial recognition methodology adopted in this work is depicted in Fig. 6. Zong et al. in [7] compare the performance of one-against-all (OAA) and one-against-one (OAO) multi-class classification using ELM. Given the multi-label dataset consisting of α different classes, OAA methodology takes into consideration α binary classifiers trained in such a way to distinguish each class and remaining classes. On the other hand, in OAO, one binary classifier is used to distinguish one pair of classes resulting in a total of $(\alpha - 1) * \alpha / 2$ binary classifiers, α being the number of different classes. As per the results presented in [7, 8] and related work using Linear Discriminant Analysis (LDA) and multiclass Support Vector Machine (SVM) [12–14], OAA has been observed to give better performance as compared to OAO methodology. Hence, for the current work, we adopt OAA ELM methodology for dealing with the facial recognition problem.

The face dataset is first split randomly into training and testing datasets as shown in Fig. 6. To each image in dataset, Viola-Jones algorithm is applied to detect the region

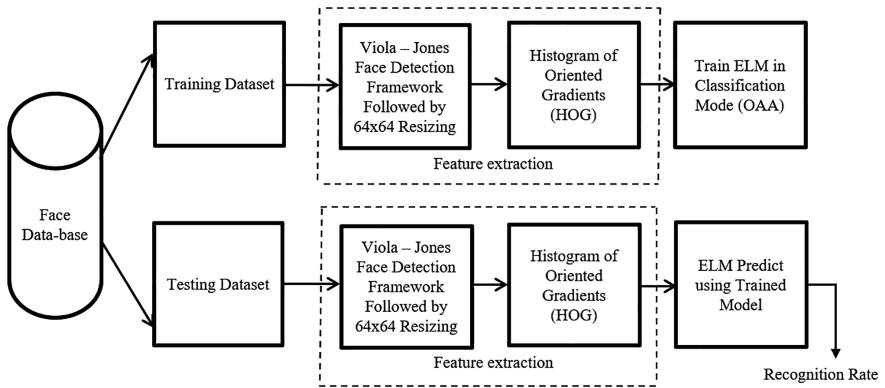


Fig. 6. Block diagram depicting the face recognition approach employing ELM in OAA mode along with Viola Jones object detection framework and HOG feature selection.

which contains useful information pertaining to a subject’s facial features. The implementation of Viola-Jones algorithm is as per the MATLAB routine given in [22]. The extracted regions have been resized to a uniform size of 64 by 64. This is done so as to ensure that the number of HOG features extracted in the subsequent stage are similar and so that the processing that follows is uniform for all the subjects under consideration. The HOG features returned by the MATLAB routine [25] is in form of a row vector and consists of 1764 features for the methodology adopted. All such row vectors corresponding to each image in training dataset is stacked one over the other to form the final dataset that would be fed to an ELM for multi-class classification as depicted in Fig. 7. Once the ELM gets trained the images in testing dataset are subjected to same processing and fed to the trained ELM model for classification. The recognition rate is then determined by obtaining the total number of correct hits to the total images under the testing dataset.

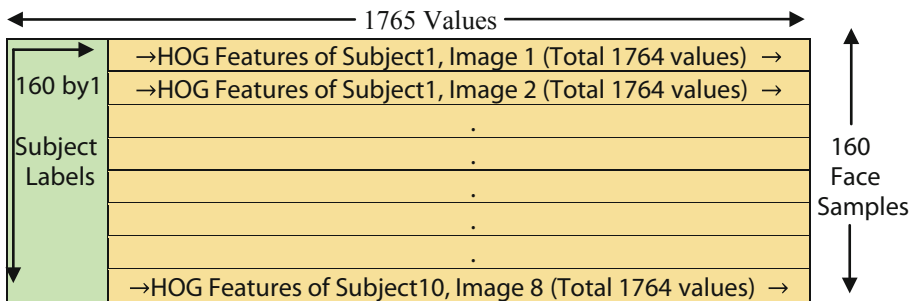


Fig. 7. Depiction of the final dataset that is to be fed to an ELM considering an AT&T database with 80:20 split, giving 160 face images for training dataset. [Note: The images are partitioned randomly into testing and training datasets. The depiction is for understanding purposes.]

6 Experimental Results and Comparisons

The facial recognition scheme so presented has been tested on standard face recognition datasets viz. AT&T [26] and YALE [27]. AT&T consists of ten different images of each 40 distinct subjects with varying lighting conditions, facial expressions and at different time instants. Each image has a dimension of 92×112 with 256 grey levels per pixel and is available in portable gray map (PGM) format. The YALE dataset on the other hand consists of 11 images for each 15 distinct individuals with different facial expressions, varying lighting conditions, and with miscellaneous eye wear. Each image in the YALE dataset has a dimension of 243×320 with 256 grey levels per pixels. These are available in graphics interchange format (GIF). Some sample face images from both the datasets is shown in Fig. 8.



Fig. 8. The first row depicts the face samples of a particular subject from AT&T database, while second and third rows depict face samples of another subject from YALE Dataset

For each dataset, the all images were split into training and testing datasets as per 80:20 and 70:30 splitting ratios. The simulations are carried out using Mathworks MATLAB 9.4 running on the Windows 10 Home Edition with an 8 GB of memory and an i5 0 7300 HQ (2.50 GHz) processor. The results are tabulated in Tables 1 and 2 and depicted in form of curves in Fig. 9.

Table 1. Consequence of hidden neurons on the recognition rate and testing time spans for AT&T dataset for 80:20 and 70:20 splitting ratios.

No. of Hidden Neurons (L)	80:20 Split			70:30 Split		
	Recognition Rate (%)	Training Time (ms)	Testing Time (ms)	Recognition Rate (%)	Training Time (ms)	Testing Time (ms)
100	81.50	15.6	125.0	77.12	15.6	291.9
150	84.12	15.6	312.5	78.79	62.5	296.9
200	84.25	15.6	62.5	77.25	15.6	421.9
250	79.75	62.5	265.6	63.54	62.5	359.4
500	89.06	140.6	421.9	88.25	140.6	656.3
1000	94.94	171.9	843.8	93.625	140.6	921.9

Table 2. Consequence of hidden neurons on the recognition rate and testing time spans for YALE dataset for 80:20 and 70:20 splitting ratios

No. of Hidden Neurons (L)	80:20 Split			70:30 Split		
	Recognition Rate (%)	Training Time (ms)	Testing Time (ms)	Recognition Rate (%)	Training Time (ms)	Testing Time (ms)
100	89.16	15.6	109.4	56.44	15.6	187.5
150	76.16	62.5	125.0	62.78	62.5	218.8
200	96.67	625	187.5	82.34	15.6	281.3
250	99.5	15.6	125.0	86.11	15.6	312.5
500	100	15.6	125.0	93.33	31.3	343.8
1000	100	78.1	187.5	96.11	93.8	406.3

The recognition rate mentioned in Tables 1 and 2 and its depiction in Fig. 9 correspond to the average recognition rate so obtained after 20 iterations. This is done to average out the error that emanates due to not so good generalization capabilities of ELM. The ELM may operate in milliseconds regime as dictated by Tables 1 and 2, but lacks in generalization due to random weights being allocated between the input/output and the hidden layers. The results so presented bring forth the fact that as the number of hidden neurons are increased, the recognition rates tend to cross the 90% marker, but the price is paid in terms of the training time spans which although increase but still remain in the scale of few milliseconds.

It is very clear from the results compiled in Tables 1 and 2 that the 80:20 splitting is better placed in comparison to 70:30 splitting ratio. The recognition rate of more than 90% and the training and testing time spans computed thereof is found to be better in case of YALE dataset. Therefore, it is suggested that the images which are captured at different orientations with varying illumination need to be stored in the GIF file format. However, in both these cases, the optimized number of hidden neurons comes out to be slightly more than 250. The recognition rate and the computed testing time first dips prior to L = 250 and then maximizes after this value. This is a pattern which is observed in the case of both datasets. The testing time is observed to be inversely varying the recognition rate. At around L = 250, the testing times (milliseconds) are maximum which then dip to a lower value in case of both datasets. Therefore, according to us, the optimized value of L = 250 for which both recognition rate (%) and testing time (milliseconds) are better placed in YALE dataset in comparison to AT&T dataset.

In order to evaluate the performance of the facial recognition scheme so presented, we compare the results with some state-of-the-art methods for different datasets under Tables 3 and 4.

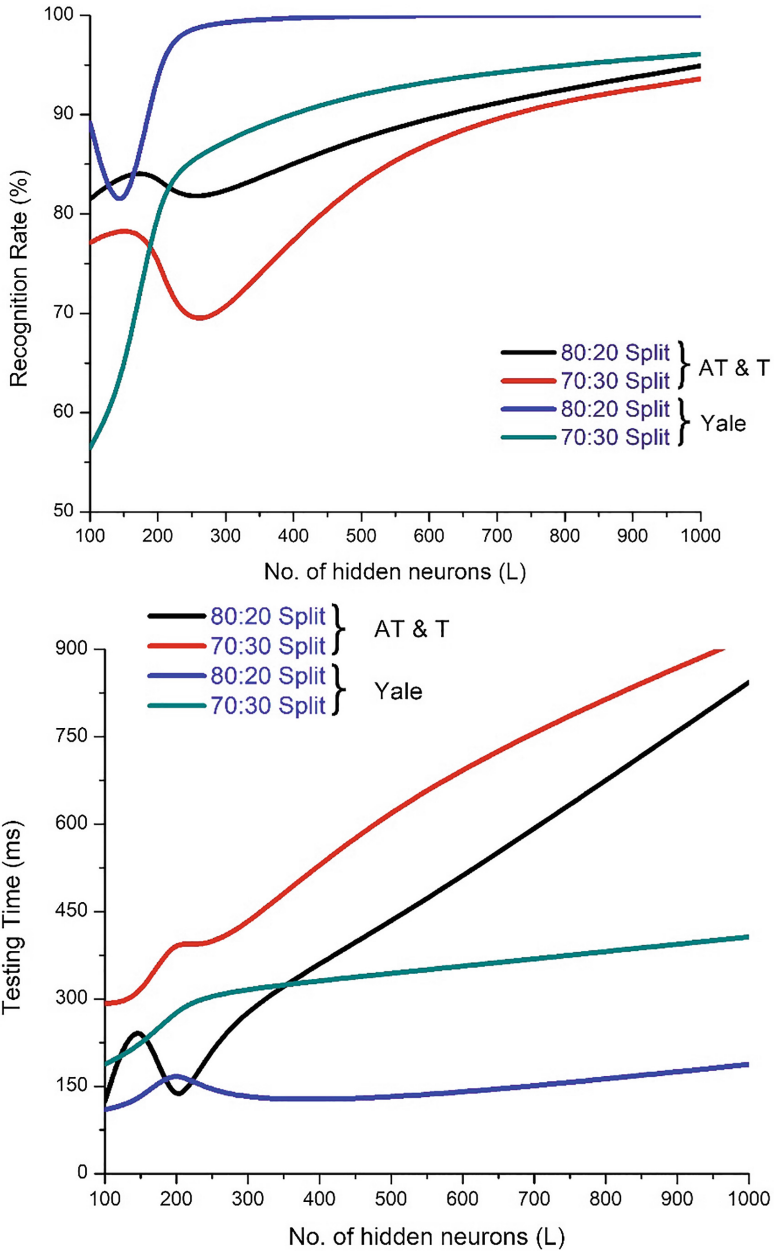


Fig. 9. Curves depicting the effect of hidden neurons on recognition rate, and testing time for the two datasets under 80:20 and 70:30 splitting ratios

Table 3. Comparison of recognition rate and running time spans for some state-of-the art facial recognition techniques for AT&T dataset taking 1000 Hidden Neurons

AT&T Dataset (70:30 Split)	Recognition Rate(%)	Training Time(ms)
PCA – NN [7]	91.67	-
LDA – NN [7]	95.50	-
DLA – NN [7]	97.62	-
PCA – ELM [7]	97.11	6380
LDA – ELM [7]	96.46	7680
DLA – ELM [7]	98.44	4540
SPSO – ELM [11]	93.57	102 x 10 ³
Fuzzy – HOG [15]	92.60	-
LDB – HOG [16]	92.00	-
Proposed Work 32 x 32 Resizing	95.00	109.4
Proposed Work 64 x 64 Resizing	93.63	140.6

Table 4. Comparison of recognition rate and running time spans for some state-of-the art facial recognition techniques for YALE dataset taking 1000 hidden neurons

YALE Dataset (70:30 Split)	Recognition Rate (%)	Training Time (ms)
PCA – NN [7]	60.50	-
LDA – NN [7]	83.83	-
DLA – NN [7]	87.25	-
PCA – KELM [8]	85.58	39
LDA – KELM [8]	85.92	23
PCA – ELM [7]	82.08	2210
LDA – ELM [7]	83.75	1240
DLA – ELM [7]	89.04	1680
LDB – HOG [16]	92.00	-
Proposed Work 40 x 40 Resizing	93.33	15.6
Proposed Work 64 x 64 Resizing	96.11	93.8

A close observation of the data compiled in Tables 3 and 4 yields a similar pattern. Clearly, our results are better placed than the existing methods presented in this paper particularly for the YALE dataset. This is primarily due to the use of GIF image file format. Additionally, the computed testing time spans are also found to be better than the ones reported by other research groups. We, therefore, conclude that our facial recognition technique not only gives better results in terms of the recognition rate (%), but our testing time span is also measured in milliseconds domain thereby suggesting that all necessary procedures-pre-processing, feature extraction and classification of images is carried out in real-time. This is possible only due to the use of a combination of several existing algorithms in this work. The combination is that of the Viola-Jones Algorithm for object identification, HOG based feature selection and the use of Extreme Learning Machine (ELM) for pattern classification. This combination brings in the desirable novelty of the proposed facial recognition technique.

7 Conclusions

A novel facial recognition technique working in real-time domain is proposed in this work. The technique involves the use of existing Viola-Jones algorithm for object identification, the Histogram of Oriented Gradients (HOG) based feature extraction and a single layer feed-forward neural network commonly known as Extreme Learning Machine (ELM). Two different datasets of images are considered for this work. These are AT&T and YALE which have several hundred images in different orientations with varying illumination levels. The ELM is found to carry out successful classification in both the datasets. Our technique, however, gives better results in case of YALE dataset as compared to other similar techniques reported in this paper. We conclude that the better results so obtained are primarily due to the GIF image file format used in YALE and due to the fast processing carried out by Viola-Jones algorithm with HOG feature selection procedures. The extremely fast classification (in milliseconds time domain) carried out by the ELM further supplements it. Overall, we find that a very high recognition rate (%) is achieved in the milliseconds time scale. We therefore conclude that the proposed facial recognition technique outperforms several other similar schemes more particularly for the YALE dataset.

Acknowledgments. The authors would like to thank University School of Information, Communication and Technology, Guru Gobind Singh Indraprastha University and Deen Dayal Upadhyaya College, University of Delhi for providing the necessary software and infrastructure support. The authors also acknowledge Faculty of ESTEM, University of Canberra for providing the necessary financial support.

References

1. Zhao, W., Chellappa, R., Philips, P.J., Rosenfeld, A.: Face recognition: a literature survey. *ACM Comput. Surv. (CSUR)* **35**(4), 399–458 (2003)
2. Turk, M., Pentland, A.: Eigenfaces for recognition. *J. Cogn. Neurosci.* **3**(1), 71–86 (1992). Massachusetts Institute of Technology
3. Gumus, E., Kilic, N., Sertbas, A., Ucan, O.N.: Evaluation of face recognition techniques using PCA, wavelets and SVM. *Expert Syst. Appl.* **37**(9), 6404–6408 (2010)
4. Li, S.Z., Jain, A.K.: *Handbook of Face Recognition*. Springer, London (2005). <https://doi.org/10.1007/978-0-85729-932-1>. ISBN 0-387-40595-X
5. Chude-Olisah, C.C., Sulong, G., Chude-Onkonkwo, U.A., Hashim, S.Z.: Illumination normalization for edge-based face recognition using the fusion of RGB normalization and gamma correction. In: *IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, pp. 412–416 (2013)
6. Du, S., Ward, R.: Wavelet-based illumination normalization for face recognition. In: *IEEE International Conference on Image Processing (ICIP)*, vol. 2, pp. 954–956 (2005)
7. Zong, W., Huang, G.B.: Face recognition based on extreme learning machine. *Neurocomputing.* **74**(16), 2541–2551 (2011)
8. Zong, W., Zhou, H., Huang, G.-B., Lin, Z.: Face recognition based on kernelized extreme learning machine. In: Kamel, M., Karray, F., Gueaieb, W., Khamis, A. (eds.) *AIS 2011. LNCS (LNAI)*, vol. 6752, pp. 263–272. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-21538-4_26

9. Iosifidis, A., Tefas, A., Pitas, I.: Enhancing ELM-based facial image classification by exploiting multiple facial views. In: *Procedia Computer Science: International Conference on Computational Science*, vol. 51, pp. 2814–2821. Elsevier (2015)
10. Rujirakul, K., So-In, C.: Histogram equalized deep PCA with ELM classification for expressive face recognition. In: *IEEE International Workshop on Advanced Image Technology* (2018)
11. Wang, Y., Li, H., Guo, Y.: Face recognition based on ICA and SPSO-ELM. In: *IEEE Information Technology, Networking, Electronic and Automation Control Conference*, pp. 602–606 (2018)
12. Zhang, G.Y., Peng, S.Y., Li, H.M.: Combination of dual-tree complex wavelet and SVM for face recognition. In: *Proceedings of International Conference on Machine Learning and Cybernetics*, vol. 5, pp. 2815–2819 (2008)
13. Gan, J.Y., He, S.B.: Face recognition based on 2DLDA and support vector machine. In: *Proceedings of International Conference on Wavelet Analysis and Pattern Recognition*, pp. 211–214 (2009)
14. Zhao, L., Song, Y., Zhu, Y., Zhang, C., Zheng, Y.: Face recognition based on multiclass SVM. In: *Proceedings of Chinese Control and Decision Conference*, pp. 5871–5873 (2009)
15. Salhi, A.I., Kardouchi, M., Belacel, M.: Histograms of fuzzy oriented gradients for face recognition. In: *IEEE International Conference on Computer Applications Technology* (2013)
16. Wang, H., Zhang, D., Miao, Z.: Fusion of LDB and HOG for face recognition. In: *IEEE 37th Chinese Control Conference*, pp. 9192–9196 (2018)
17. Huang, G.B., Zhu, Q.Y., Siew, C.K.: Extreme learning machine: theory and applications. *Neurocomputing* **70**, 489–501 (2006). Elsevier
18. Huang, G.B., Zhu, Q.Y., Siew, C.K.: Real-time learning capability of neural networks. *IEEE Trans. Neural Netw.* **17**(4), 863–878 (2006)
19. Huang, G.B.: The MATLAB code for ELM (2004). <http://www.ntu.edu.sg/home/egbhuang>
20. Viola, P., Jones, M.J.: Robust real-time face detection. *Int. J. Comput. Vis.* **57**(2), 137–154 (2004)
21. Lo, C., Chow, P.: A high-performance architecture for training Viola-Jones object detectors. In: *IEEE International Conference on Field-Programmable Technology*, pp. 174–181 (2012)
22. Mathworks MATLAB: Detection objects using the Viola-Jones algorithm. Mathworks MATLAB Documentation R2018b (2012). <https://in.mathworks.com/help/vision/ref/vision.cascadeobjectdetector-system-object.html>
23. Dalal, N., Triggs, B.: Histogram of oriented gradients for human detection. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 886–893 (2005)
24. Korkmaz, S.A., Akçiçek, A., Bínol H., Korkmaz, M.F.: Recognition of the stomach cancer images with probabilistic HOG feature vector histograms by using HOG features. In: *IEEE International Symposium on Intelligent Systems and Informatics (SISY)*, pp. 339–342 (2017)
25. Mathworks MATLAB: Extract histogram of oriented gradients (HOG) features. Mathworks Matlab Documentation R2018b (2013). https://in.mathworks.com/help/vision/ref/extractthogfeatures.html?s_tid=doc_ta
26. AT&T Laboratories Cambridge: The AT&T Dataset (formerly ‘The ORL Dataset of Faces’). http://www.cl.cam.ac.uk/Research/DTG/attarchive:pub/data/att_faces.zip
27. YALE Face Dataset. <http://cvc.cs.YALE.edu/cvc/projects/YALEfaces/YALEfaces.html>