A Novel Non-Iterative Deep Convolutional Neural Network with Kernelized Classification for Robust Face Recognition

Virendra P. Vishwakarma
USIC&T, Guru Gobind Singh Indraprastha University, India
vpv@ipu.ac.in

Reena Gupta
USIC&T, Guru Gobind Singh Indraprastha University, India
reena@ipu.ac.in (corresponding author)

Abhay Kumar Yadav
USIC&T, Guru Gobind Singh Indraprastha University, India
abhaybbdnitm01@gmail.com

Received: 26 June 2024 | Revised: 18 July 2024 | Accepted: 26 July 2024

 DOI: https://doi.org/10.48084/etasr.8229

ABSTRACT

Deep Convolutional Neural Networks (DCNNs) are very useful for image-based pattern classification problems because of their efficient feature extraction capabilities. Although DCNNs have good generalization performance, their applicability is limited due to slow learning speed, as they are based on iterative weight-update algorithms. This study presents a new noniterative DCNN that can be trained in real-time. The fundamental block of the proposed DCNN is fixed real number-based filters for convolution operations for multi-feature extraction. After a finite number of feature extraction layers, nonlinear kernel mapping along with pseudo-inverse is used for the classification of extracted feature vectors. The proposed DCNN, named Deep Convolutional Kernelized Classification (DCKC), is noniterative, as the mask coefficients of its convolution operations are fixed real numbers. The kernel function with predefined parameters of DCKC does a nonlinear mapping of extracted features, and pseudo-inverse is used to find its output weights. The proposed noniterative DCKC was evaluated on benchmark face recognition databases, achieving better results and establishing its superiority.

Keywords-convolutional neural networks; deep neural networks; non-iterative learning; face recognition

I. INTRODUCTION

The real challenge in machine learning is to adequately simulate the human brain for efficient information processing. Deep Neural Networks (DNN) are better models of how the human brain works than shallow neural networks (with one or two hidden layers). Two Dimensional Convolution operation-based DNNs (DCNNs) are one of the most promising models. DCNNs have been increasingly used in image and speech recognition problems and have performed well in a variety of tasks [1]. Facial Recognition (FR) is one of the most popular domains of DCNN application [2] and is becoming more successful as a technology that is easy to use and unobtrusive. The appearance, expression, and light varieties affect the results of classification frameworks [3]. Light variation on subject faces arises due to light projection from distinctive bearings at various points on the face of an individual, creating pictures that are clearly distinctive, making the recognition process exceptionally troublesome. Several studies have presented strategies for light normalization.

DCNNs were initially introduced in 2010 [4], exploring their application in computer vision and handwritten digit recognition. The excellent performance of DCNNs over traditional Computer Vision (CV) classification methods has attracted the interest of researchers over the years [5, 6]. DCNNs have revolutionized the field of CV by providing a comprehensive approach. In 2014, DeepFace was introduced [7], which is an FR system that achieved remarkable accuracy on the LFW benchmark, approaching human performance. However, subsequent systems, such as DeepId3 and FaceNet [8], quickly exceeded its accuracy. These advances in FR technology are a testament to the advancement in deep-network architectures. The evolution of DCNNs can be traced back to LeNet in 1989 [9], and since then have become sophisticated networks driven by challenges such as the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [10]. Among the influential ILSVRC networks, LeNet, GoogleNet, AlexNet, VGGNet, ZFNet, and ResNet [11] stand out, as shown in Figure 1. A typical DCNN follows a conventional structure, which includes a series of convolutional layers, contrast normalization, and max-pooling, along with one or more fully connected layers. In addition, various modifications have been implemented to enhance performance.

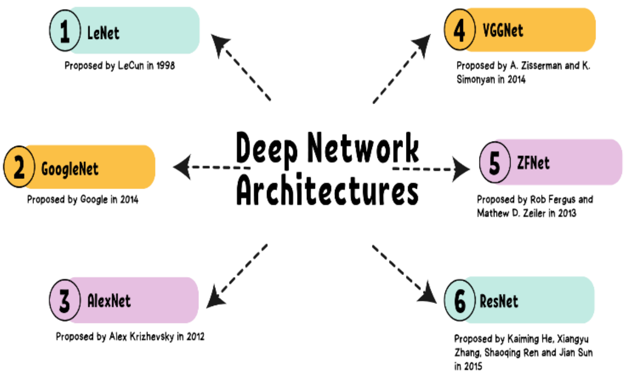

Fig. 1. Progress in deep learning architectures.

The advance of deep-network architectures began by enlarging both their depth and width. However, the growing complexity associated with larger networks is not preferred in practical situations. As a result, GoogleNet was introduced with an enhanced architecture along with a few feature parameters. Subsequently, Microsoft aimed to streamline training using networks with reduced complexity. In recent developments, these two design techniques have been merged to produce more simplified networks [12].

An efficient DCNN learning algorithm will reduce training time and achieve higher accuracy. Existing state-of-the-art DCNNs, such as AlexNet, GoogleNet, ResNet, etc., use backpropagation to calculate the parameters of the network. The backpropagation algorithm is a stochastic iterative algorithm based on the steepest-decent theory, where the weights of the neural network are updated to negative gradients in the weight space. Incorrect learning rate, under-fitting, over-fitting, local minima, slow learning, and out-of-memory problems are some basic implementation issues with existing algorithms [13]. This study aims to develop a new classification engine that can extract discrimination-effective features to efficiently classify the nonlinear and nonconvex space of facial images captured under unbounded constraints. The objective is to achieve multiple feature extraction and noniterative learning. The highlights of the proposed method are:

- A deterministic learning method is proposed to extend the single-hidden-layer noniterative classifiers. In existing noniterative classifiers, such as Extreme-Learning-Machine (ELM), the input parameters are assigned randomly and the number of neurons in the hidden layer is obtained experimentally. In the proposed method, the parameters are fixed, resulting in a stable and invariant output.
- Multiple features are extracted with the help of conglomerate convolution filters, where the filter coefficients are fixed without requiring iterative tuning of the coefficients. In existing DCNNs, these coefficients are tuned iteratively. In a non-iterative classifier, Kernel-Extreme-Learning-Machine (KELM), multiple layers for feature extraction do not exist.
- To deal with the real-time challenges of FR systems, the low objects and the Low-Frequency Discrete Cosine Transform (LFDCT) components are adjusted to overcome the effects of lighting conditions with the help of a fuzzy filter-based illumination-normalization method.
- To establish the effectiveness of the proposed method, it was compared with standard noniterative FR methods, showing promising results that demonstrate its competence.

II. DCNN PRELIMINARIES

The output matrix of the hidden layer can be expressed as:

$$Z = \sum_{r=1}^{R} \tau_r \beta(a_r, b_r, x) = \tau . H(x) \quad (1)$$

with a_r and b_r being the input weights along with the bias of the r^{th} hidden layer neuron with a random number. $H(x) = [\beta(a_1, b_1, x) \dots, \beta(a_R, b_R, x)]$ is the output matrix obtained from the hidden layer for input x. For a single-hidden-layer feedforward NN, the output weight τ is mathematically obtained. The connection of output neurons with hidden nodes can be obtained by:

$$\tau = H^{-1} Z = H^T \left(\frac{I}{C} + HH^T\right)^{b-1} T \quad (2)$$

where T is the target class and C is the regularization coefficient. ELM was introduced to minimize training errors and the norm of the output weights, denoted as $\|\tau\|^2$. Compared to gradient-descent methods, ELM offers a significant advantage in terms of time efficiency [14]. It also allows for a reduction in computational time required for parameter optimization. However, ELM suffers from nondeterministic performance due to the random weights and biases it employs. To address the limitations of ELM, a noniterative algorithm based on the kernel matrix was proposed. In KELM, the kernel matrix is not directly related to the target class but is only relevant to training samples [15]. The kernel matrix can be mathematically expressed as:

$$\varphi = \sum_{e=1}^{R} H(a_e, b_e, x_t) . H(a_e, b_e, x_t) \quad (3)$$

III. PROPOSED NONITERATIVE DCNN WITH KERNELIZED CLASSIFICATION

As the FR problem under real-world variations (pose, illumination, expression, etc.) is highly curved and nonlinear, it can be resolved by either efficient feature mining, enhanced classification system, or integrating both. The proposed DCKC comprises some layers for feature extraction followed by noniterative kernel mapping along with pseudo inverse for classification. The feature extraction component is highly responsible for the performance of any classification system. Here, multiple convolution operations are performed to extract features using fixed-valued masks. As shown in Figure 2, each layer comprises some convolution filters, followed by a Rectified Linear Activation Function (RLAF) and a pooling operation. These three operations together, named CRP, operate in multiple layers on each input image. After the *N*-layer operation of the CRP, the multidimensional outcome is converted into a column feature vector. This high-dimensional feature vector is fed to the kernel function for non-linear mapping, followed by pseudo-inverse for classification.

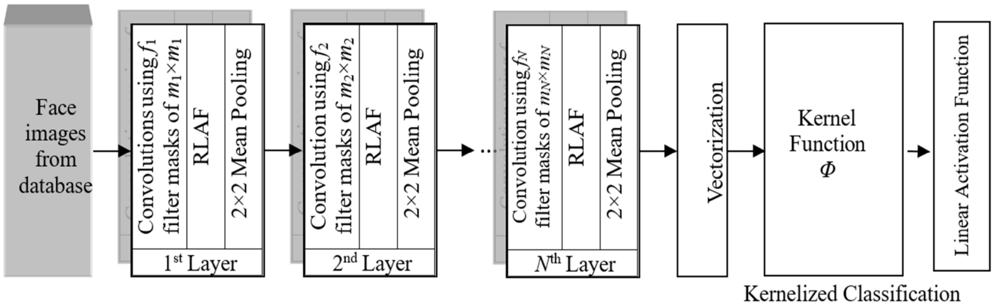

Fig. 2. Proposed DL architecture.

A. *Convolution Using Fixed-Valued Masks*

As shown in Figure 2, all input images are convolved using f_1 masks in the first layer of the proposed NN. The size and number of these masks vary from layer to layer, that is, f_i masks with size $m_i \times m_i$ are used for convolution in the i^{th} NN layer. If w_j is a particular mask, then the convolution is defined as:

$$c(x,y) = m \sum_{s=-m_i/2}^{m_i/2} \sum_{t=-m_i/2}^{m_i/2} w_j(s,t) d(x-s, y-t) \quad (4)$$

where *d* is two-dimensional input data, *c* is convolved data, and w_j is the j^{th} convolution mask of size $m_i \times m_i$. The elements of the masks are generated using the MATLAB function *randn* [16], which gives normally distributed random numbers. To repeat the same set of elements that is the same mask matrices (fixed), the state of the *randn* function is controlled by the function *rng*, which controls the generation of random numbers depending upon the value of its argument (seed). *randn* along with *rng*(1) generates the set of real numbers that are repeatable and deterministic. For example, *rng*(1),*randn*(4,4,3) gives a set of three masks of size 4×4, which will be the same matrices whenever the code is executed. These matrices are given in Table I.

TABLE I. NORMAL DISTRIBUTED FIXED NUMBERS

-0.65	-0.85	-0.20	-1.51
1.18	-0.57	0.59	0.88
-0.76	-0.56	-0.85	-0.24
-1.11	0.18	0.80	0.17

-1.97	-0.28	-1.87	-1.37
-1.27	0.60	-1.05	-0.29
1.18	1.78	-0.42	1.27
2.03	1.77	1.40	0.07

0.45	-0.49	0.60	-1.04
-0.32	1.80	-0.54	-0.35
0.79	0.59	-0.16	-1.17
0.93	-0.64	0.61	-0.69

All elements in the convolution filter masks in every layer are multiplied by 10^{-4}, the value of multiplier *m* in (4), to keep the convolved output small. This is performed considering that the weights of the DCNN should be small because the values of the synaptic weights in a neural network are more dominant than its size. A neural network with a smaller norm of synaptic weights gives smaller training errors and better generalization performance.

B. *Rectified Linear Activation Function (RLAF)*

All the convolution output matrices are operated by an activation function. In the proposed method, RLAF is used due to its simplicity. It simply outputs the given input, when the input is positive. For negative input values, the RLAF output is zero. Mathematically, the output of RLAF *y* can be expressed as:

$$y = max(0, x) \quad (5)$$

where *x* represents the output of (4).

C. *Pooling*

At each layer, a 2×2 mean pooling operation is performed on the output of RLAF. For the j^{th} block R_j (of size 2×2), the output of mean pooling can be expressed as:

$$p_j = \frac{1}{|R|} \sum_{i \in R_j} y_i \quad (6)$$

D. *Kernelized Classification*

To train DCKC, a set of input specimens $\{p_k, t_k\}_{k=1}^{Q}$ is presented, where Q is the total number of specimens. $p_k \in R^{M \times N}$ denotes the k^{th} training face image, and t^k represents the corresponding class vector.

As previously stated, FR under uncontrolled environments, such as varying illumination, pose, expression, etc., is extremely nonlinear and nonconvex, degrading the accuracy of many contemporary FR methods. The proposed DCKC overcomes these problems by extracting multiple features with the help of the convolution operation. It further enhances the feature extraction capability using the kernel function [17] to map the given input features, obtained from CRP operations, into a kernel feature space. This is performed to construct a better feature space (kernel feature space) that is less nonlinear and less nonconvex than that in the input feature space of the kernel function. If there are two feature vectors, say x_i and x_j, then the output corresponding to these inputs using a kernel function Φ can be obtained as:

$$\chi_{i,j} = \Phi(x_i, x_j); i, j = 1, 2, \dots, Q \quad (7)$$

Many kernel function types can be used, including polynomial, Laplacian, sigmoid, wavelet, and Radial Basis Function (RBF) kernels. In case where the activation function of the output layer neurons is linear and no biases are applied to these neurons, the output of the SLFN with this architecture can be expressed as:

$$\sum_{i=1}^{Q} \eta_i \varphi(iw_i^T . x_j + b_i) = a_j, \text{ for } j = 1, 2, \dots, Q \quad (8)$$

The training sample represented in (6) can be learned completely by the proposed architecture represented in Figure 2, without any error, if there are η_i, iw_i, and b_i such that:

$$\sum_{i=1}^{Q} \eta_i \varphi(iw_i^T . x_j + b_i) = \boldsymbol{t}_j, \text{ for } j = 1, 2, \dots, Q \quad (9)$$

The above equations can be expressed as:

$$H\Gamma = T \quad (10)$$

where

$$H = \begin{bmatrix} \varphi(iw_1^T . x_1 + b_1) & \dots \varphi(iw_Q^T . x_1 + b_Q) \\ \vdots & \vdots \\ \varphi(iw_1^T . x_Q + b_1) & \dots \varphi(iw_Q^T . x_Q + b_Q) \end{bmatrix} \quad (11)$$

$$\Gamma = \begin{bmatrix} \eta_1^T \\ \vdots \\ \eta_Q^T \end{bmatrix} \text{ and } = \begin{bmatrix} t_1^T \\ \vdots \\ t_Q^T \end{bmatrix} \quad (12)$$

IV. EXPERIMENTAL RESULTS AND ANALYSIS

All experiments were carried out in MATLAB 2024a and executed on a laptop with an Intel i7 2.9 GHz processor with 16 GB RAM. It is worth mentioning that the proposed DCKC multi-layer deep neural network classifier did not need a GPU or cloud environment to execute the experiments. Three different publicly available facial image databases ATT [18], Yale [19], and AR [20] were used. Figure 3 shows a detailed description of the databases.

Tables II, III, and IV show the performance of the proposed method, in terms of percentage test error, in these three databases. There are variations in the percentage test error rate, which refers to the number of misclassifications for the total number of samples in the test dataset. This analysis was performed for varying numbers of training images per subject (NTIPS). For example, in the Yale database, when one individual image per subject was used for training, the remaining images of the subject were used for testing. Similarly, when two images per subject were used for training, the remaining images of the subject were used for testing. The selection of these images was performed sequentially to form sets for training and test images.

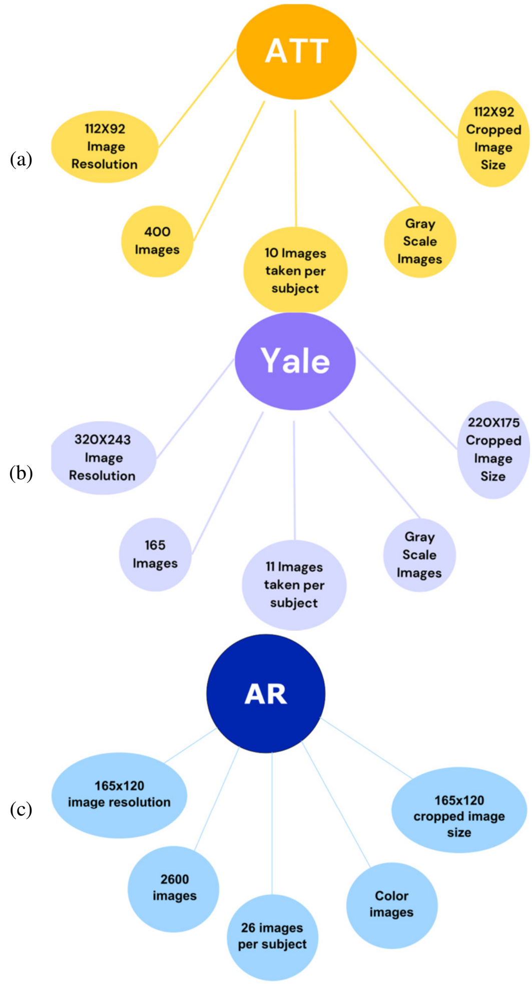

Fig. 3. Databases used.

Polynomial, Laplacian, sigmoid, wavelet, and RBF kernels were chosen along with parameter exploration. As multiple convolution filters are applied for global feature exploitation, according to the literature, RBF kernels are a better choice for local feature wrenching. The RBF kernel parameter σ was determined to be 800 after a comprehensive set of experiments. This value of the RBF parameter was used for all the results shown.

As shown in Table II, the percentage error rate was zero in the ATT database, demonstrating the classification performance of the proposed method. Accurate classification was achieved for 70, 80, and 90% of the training data (correspondingly 30, 20, and 10% of the test data) using the proposed method in the ATT face database. Similarly, on the Yale and AR face databases, significantly better performance was achieved. As face images in the Yale and AR databases are affected by light variations, fuzzy filter-based illumination normalization [17] was applied to compensate for the effect of light variations. After a thorough investigation, the best performance was obtained for an LFDCT of 91. Performance deteriorated after increasing LFDCT. Similarly, results in the AR database were obtained for an LFDCT of 105.

The total number of face images is 26 per subject in the AR face database. In the performance analysis on this database, 1, 2, …, 6, and 13 images per subject were used to form the distinct training sets, and the exclusive remaining images were used to form the test sets. It was found that even with 50% of images used for the test set, the proposed method achieved accurate classification, as shown in Table IV.

TABLE II. PERCENTAGE TEST ERROR OF THE PROPOSED METHOD ON THE ATT DATABASE

Sets	Number of images for training/test sets					
Training	1	2	3	4	5	6
Testing	9	8	7	6	5	4
% test error	15.56	4.69	1.43	1.25	1.0	1.25

TABLE III. PERCENTAGE TEST ERROR OF THE PROPOSED METHOD ON THE YALE DATABASE

Sets	Number of images for training/test sets					
Training	1	2	3	4	5	6
Testing	10	9	8	7	6	5
Output	6.67	0	0	0	1.11	0

TABLE IV. PERCENTAGE TEST ERROR OF THE PROPOSED METHOD ON THE AR DATABASE

Sets	Number of images for training/test sets						
Training	1	2	3	4	5	6	13
Testing	25	24	23	22	21	20	13
Output	24.240	13.500	6.478	5.272	4.809	1.35	0

A. Computational Complexity Analysis

Q is used to denote the number of training samples, the size of input images is represented as $d\times d$, h denotes the number of hidden layer neurons, and w denotes the number of classes of the database. The proposed method comprises the following processing steps:

1. The time complexity of the convolution operation is $O(d \log d)$. The RLAF operation is of linear time complexity and the time complexity of mean pooling is the same as that of the convolution operation. As there is a fixed number of CRP operations, the total time complexity of feature extraction of the proposed method is $O(d \log d)$.
2. The time complexity of the kernel matrix (7) is $O(Q^2 f)$, where f is the size of the feature vector after a finite number of CRP layers.
3. The time complexity of finding the output weight Γ(9) and (11) using pseudo-inverse is $O(Q^3 + Q^2 w)$.

From the above, it is concluded that the total time complexity of the proposed method is $O(d \log d + Q^2 f + Q^3 + Q^2 w)$. As the size of the input image d is comparatively smaller than the number of training samples Q, the time complexity of the proposed method can be approximated to $O(Q^3)$, which is the same as that of the most promising existing noniterative classifier DMK-ELM-FFE [14].

V. COMPARATIVE ANALYSIS

Tables V, VI, and VII show the performance comparison of the proposed method with existing noniterative state-of-the-art methods on these databases. The accuracy achieved using the proposed method is significantly higher.

TABLE V. ACCURACY-BASED COMPARATIVE PERFORMANCE EVALUATION ON THE ATT DATABASE

Technique	Number of sample training images taken per subject					
	1	2	3	4	5	6
ELM [21]	62.500	74.690	78.570	83.500	83.500	90.630
KELM RBF[22]	67.220	77.810	80.360	87.080	89.500	95.000
LBP-KELM[23]	67.720	78.100	81.610	87.480	89.190	95.000
OMKELM[24]	68.330	79.370	85	91.250	92	95.000
DMK-ELM[14]	73.890	86.560	85.710	91.250	92.500	95.370
DMK-ELM-FFE [14]	75.280	89.320	93.580	95.830	98.500	96.810
Proposed DCKC	**84.440**	**95.310**	**98.570**	**98.750**	**99**	**98.75**

TABLE VI. ACCURACY-BASED COMPARATIVE PERFORMANCE EVALUATION ON THE YALE DATABASE

Technique	Number of sample training images taken per subject					
	1	2	3	4	5	6
ELM [21]	54.100	82.410	85.350	89.520	88.800	94.000
KELM RBF[22]	62.670	87.410	90	91.430	93.330	96.000
LBP-KELM[23]	47.980	61.770	79.250	87.610	89.310	88.830
OMKELM[24]	64.670	88.150	92.500	93.330	94.440	96.000
DMK-ELM[14]	65.330	89.630	95	96.190	95.560	98.670
DMK-ELM-FFE[14]	65.330	91.890	95.560	96.190	97.500	98.330
Proposed DCKC	**93.333**	**100**	**100**	**100**	**98.889**	**100**

TABLE VII. ACCURACY-BASED COMPARATIVE PERFORMANCE EVALUATION ON THE AR DATABASE

Technique	Number of sample training images taken per subject					
	1	2	3	4	5	6
ELM [21]	67.540	66.420	68.640	70.500	74.220	71.500
KELM RBF[22]	60.310	62.000	64.910	74.000	80.330	81.130
LBP-KELM[23]	70.850	74.750	76.360	83.800	91.440	93.250
OMKELM[24]	70.850	74.750	76.360	83.800	91.440	93.250
DMK-ELM[14]	77.690	82.330	84.550	87.700	93.440	94.630
DMK-ELM-FFE[14]	78.080	82.250	84.270	91.000	95.440	96.750
Proposed DCKC	75.760	86.500	93.521	94.727	95.196	98.65

VI. CONCLUSIONS

This study presented a novel method for face image classification under uncontrolled variations in illumination, pose, and expressions using the principle of convolution filter-based feature extraction and noniterative learning. Although existing DCNNs have notable generalization performance, their application is limited due to their slow learning speed in real-time environments. This study presented a new noniterative DCNN that can learn in real-time. The main component of the proposed DCNN is fixed real number-based filters for convolution operation for multi-feature extraction. After the limited number of feature extraction layers, nonlinear kernel mapping with pseudo-inverse is used to classify the selected feature vectors. The proposed DCKC is noniterative because the mask coefficients of its convolution operations are fixed real numbers. The kernel function with predefined DCKC parameters is a nonlinear mapping of the extracted features, and the pseudo-inverse is used to find the DCKC output weight. The proposed DCKC was tested on benchmark face recognition databases. Significantly better results were obtained compared to those of other approaches, indicating the superiority of the proposed method. A limitation of this study is that the sensitivity of the model to various hyperparameters was not extensively explored. A detailed analysis could provide insights into the stability and robustness of the model across different settings. In future endeavors, the proposed DCKC will be explored for video-sequence-based person identification. In addition, incorporating other components of soft computing, such as fuzzy logic, will be explored to cater to the ambiguity of inter- and intra-person class dependencies.

ACKNOWLEDGMENT

The authors express their gratitude to Guru Gobind Singh Indraprastha University, New Delhi, India.

REFERENCES

[1] Y. Said and Y. A. Alsariera, "Hardware Implementation of a Deep Learning-based Model for Image Quality Assessment," *Engineering, Technology & Applied Science Research*, vol. 14, no. 3, pp. 13815–13821, Jun. 2024, https://doi.org/10.48084/etasr.7194.

[2] P. M. G. I. Reis and R. O. Ribeiro, "A forensic evaluation method for DeepFake detection using DCNN-based facial similarity scores," *Forensic Science International*, vol. 358, May 2024, Art. no. 111747, https://doi.org/10.1016/j.forsciint.2023.111747.

[3] S. Veerashetty, Virupakshappa, and Ambika, "Face recognition with illumination, scale and rotation invariance using multiblock LTP-GLCM descriptor and adaptive ANN," *International Journal of System Assurance Engineering and Management*, vol. 15, no. 1, pp. 174–187, Jan. 2024, https://doi.org/10.1007/s13198-022-01688-0.

[4] Y. LeCun, K. Kavukcuoglu, and C. Farabet, "Convolutional networks and applications in vision," in *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, Feb. 2010, pp. 253–256, https://doi.org/10.1109/ISCAS.2010.5537907.

[5] M. Bhalekar and M. Bedekar, "D-CNN: A New Model for Generating Image Captions with Text Extraction Using Deep Learning for Visually Challenged Individuals," *Engineering, Technology & Applied Science Research*, vol. 12, no. 2, pp. 8366–8373, Apr. 2022, https://doi.org/ 10.48084/etasr.4772.

[6] S. Chopparapu, G. Chopparapu, and D. Vasagiri, "Enhancing Visual Perception in Real-Time: A Deep Reinforcement Learning Approach to Image Quality Improvement," *Engineering, Technology & Applied Science Research*, vol. 14, no. 3, pp. 14725–14731, Jun. 2024, https://doi.org/10.48084/etasr.7500.

[7] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "DeepFace: Closing the Gap to Human-Level Performance in Face Verification," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, USA, Jun. 2014, pp. 1701–1708, https://doi.org/ 10.1109/CVPR.2014.220.

[8] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2015, pp. 815–823, https://doi.org/10.1109/CVPR.2015.7298682.

[9] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Aug. 1998, https://doi.org/10.1109/ 5.726791.

[10] O. Russakovsky *et al.*, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, Dec. 2015, https://doi.org/10.1007/s11263-015-0816-y.

[11] S. Patel, "A Comprehensive Analysis of Convolutional Neural Network Models," *International Journal of Advanced Science and Technology*, vol. 29, pp. 771–777, 2020.

[12] X. Du, M. El-Khamy, J. Lee, and L. Davis, "Fused DNN: A Deep Neural Network Fusion Approach to Fast and Robust Pedestrian Detection," in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Santa Rosa, CA, USA, Mar. 2017, pp. 953–961, https://doi.org/10.1109/WACV.2017.111.

[13] Y. Luo, C. Wu, and Y. Zhang, "Facial expression recognition based on fusion feature of PCA and LBP with SVM," *Optik - International Journal for Light and Electron Optics*, vol. 124, no. 17, pp. 2767–2770, Sep. 2013, https://doi.org/10.1016/j.ijleo.2012.08.040.

[14] B. Ahuja and V. P. Vishwakarma, "Deterministic multikernel extreme learning machine with fuzzy feature extraction for pattern classification," *Multimedia Tools and Applications*, vol. 80, no. 21, pp. 32423–32447, Sep. 2021, https://doi.org/10.1007/s11042-021-11097-3.

[15] A. L. Afzal, N. K. Nair, and S. Asharaf, "Deep kernel learning in extreme learning machines," *Pattern Analysis and Applications*, vol. 24, no. 1, pp. 11–19, Feb. 2021, https://doi.org/10.1007/s10044-020-00891-8.

[16] G. Lindfield and J. Penny, *Numerical Methods: Using MATLAB*. Academic Press, 2018.

[17] V. P. Vishwakarma, "Illumination normalization using fuzzy filter in DCT domain for face recognition," *International Journal of Machine Learning and Cybernetics*, vol. 6, no. 1, pp. 17–34, Feb. 2015, https://doi.org/10.1007/s13042-013-0182-4.

[18] F. S. Samaria and A. C. Harter, "Parameterisation of a stochastic model for human face identification," in *Proceedings of 1994 IEEE Workshop on Applications of Computer Vision*, Sarasota, FL, USA, Dec. 1994, pp. 138–142, https://doi.org/10.1109/ACV.1994.341300.

[19] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, Jul. 1997, https://doi.org/10.1109/34.598228.

[20] C. Zou, K. I. Kou, and Y. Wang, "Quaternion Collaborative and Sparse Representation With Application to Color Face Recognition," *IEEE Transactions on Image Processing*, vol. 25, no. 7, pp. 3287–3302, Jul. 2016, https://doi.org/10.1109/TIP.2016.2567077.

[21] W. Zong and G. B. Huang, "Face recognition based on extreme learning machine," *Neurocomputing*, vol. 74, no. 16, pp. 2541–2551, Sep. 2011, https://doi.org/10.1016/j.neucom.2010.12.041.

[22] W. Zong, H. Zhou, G. B. Huang, and Z. Lin, "Face Recognition Based on Kernelized Extreme Learning Machine," in *Autonomous and Intelligent Systems*, Burnaby, Canada, 2011, pp. 263–272, https://doi.org/10.1007/978-3-642-21538-4_26.

[23] B. Ahuja and V. P. Vishwakarma, "Local Binary Pattern Based Feature Extraction with KELM for Face Identification," in *2020 6th International Conference on Signal Processing and Communication (ICSC)*, Noida, India, Mar. 2020, pp. 91–95, https://doi.org/10.1109/ICSC48311.2020.9182760.

[24] B. Ahuja and V. P. Vishwakarma, "Optimised multikernels based extreme learning machine for face recognition," *International Journal of Applied Pattern Recognition*, vol. 5, no. 4, pp. 330–340, Jan. 2018, https://doi.org/10.1504/IJAPR.2018.097105.