

# Face Recognition across Pose using ELM Framework

Kumud Arora, Poonam Garg, and Virendra P. Vishwakarma

**Abstract**—This paper proposes the use of Extreme Learning Machine regression and classification framework to recognize face across pose. ELM regression framework is used to generate virtual frontal view from its corresponding side view. Kernel version of ELM is used for the non-linear mapping estimation between frontal and its corresponding non frontal view. Non Linear estimation using proposed KELM is found to be satisfactory even in the case of reduced image coefficients. In addition to regression framework, this paper also explored few popular face features like LBP, HOG, appearance and entropy features with ELM classification framework. Combination of appearance and entropy feature set is found to give best classification accuracy among the considered feature set. Proposed experiments are conducted on subset of FERET database with images up to 45° pose variation.

**Index Terms**—ELM (Extreme Learning Machine); Face Pose; KELM (Kernel Extreme Learning Machine).

## I. INTRODUCTION

During last few years, research for accurate and reliable face recognition has made considerable progress. Machine vision face recognition depends largely on features extracted from eye, nose and mouth regions of the input face. These discriminant features becomes occluded in case of face pose like profile pose. Recognition capability degrades further due to variability in illumination and scales. To achieve the objective of robust face recognition across poses, various approaches like: pose normalization, pose adaptive filters and pose synthesis has been proposed in the literature. All these approaches extract high dimensional features (local/global) from the images and integrate them by using learning algorithms. Extreme Learning Machine (ELM) proposed by Huang et-al. [1],[2] for single hidden layer feed- forward networks, has gained importance among various learning algorithms. Speed advantage and convex model structure of ELM is responsible for its usage across various domains. ELM maps the input features to the feature space using non-linear activation function [1]-[4]. In the mapping process of the input space  $R^D$  to a high dimensional feature space  $R^L$  (ELM Space), the dimensionality  $L$  of the ELM space is usually empirically chosen. The kernel versions of the ELM classifier have been proposed in order to avoid the time consuming estimation of the ELM space dimensionality, [3], [6],[7].

The main idea in kernel versions of the ELM [3],[6] is to escape the direct calculation of outputs of the hidden layer

neurons, and the annexation of the inherent encoding in the ELM kernel matrix. Kernel matrix is defined by  $\mathbf{K} = \Phi^T \Phi$ , where  $\Phi \in R^{L \times N}$  and refers to ELM space of  $N$  training data samples. Kernel function is also largely responsible for keeping intact the geometrical structure of the data in the kernel mapped space. If inappropriate kernel is chosen then data's discriminative ability in the feature space could reduce. Kernel function becomes effective when it is able to define similarities among train and test data set. The enactment of kernel method is heavily dependent upon the kernel function parameters and the type of kernel function [10]. Various kernel functions like Radial Basis Function, Polynomial kernel, Exponential Chi-Square kernel have been used with ELM classification framework. To improve the classification /recognition accuracy, multiple kernels (MKL) are proposed. Xie et al. [20] used MKL to fuse edge features and texture features for infrared feature face recognition. MKL can use multiple predefined set of kernels in linear or non-linear combination instead of selecting one specific kernel function. Recently Goel et al. [8] explored kernel extreme learning machine (KELM) regression framework to generate virtual frontal face image from its non-frontal counterpart.

In this paper we explored both regression and classification framework of ELM to recognize faces across the poses. Non-linear regression via kernel ELM is proposed to estimate the non-linear mapping between frontal face views from its counter-part non-frontal views for effective face recognition. This work avoids time consuming step for finding optimal value for hidden neurons by experimental tuning. Compared to linear regression, KELM regression is more efficient for virtual view generation as it considers non-linear shape of the face view. The impact of kernel function of Kernel Extreme Learning Machine in regression framework for face pose normalization is also explored. Apart from evaluation of various popular kernel functions (Radial Basis Function, Polynomial, Fractional Polynomial, Generalized Triangular function), mixed kernel function Triangular Hermite (Tri-H KELM) [22] is also evaluated. Various features like LBP, HOG, entropy and intensity features are explored with ELM classification framework. The rest of this paper is organized as follows: In section 2, we give an outline of ELM classification framework as well as ELM regression framework for pose normalization. Section 3 presents the outline of the proposed pose normalization approach. In section 4 results obtained with FERET face pose images up to 45 degree deviation are analyzed and interpreted. Finally, in section 5 we conclude the paper and highlight some of the future ideas for research.

---

Published on April 30, 2018.

K. Arora is with Inderprastha Engineering College, Ghaziabad, India. (e-mail: kumud.arora76@gmail.com)

P. Garg is with Institute of Management & Technology, Ghaziabad, India. (e-mail: pgarg@imt.edu)

V. P. Vishwakarma is with Guru Gobind Singh Inderprastha University, India. (email: virendravishwa@rediffmail.com)

## II. RELATED WORK

### A. Face Recognition across Pose

Face recognition across pose talk about the matching of face images in diverse poses generally different from the enrolled one by the computers. During pose change, issues like feature self-occlusion and loss of semantic correspondence for facial texture arise. Direct matching of two faces under different poses may lead to false match or false rejection for the input test image. Out of plethora of face recognition strategies (pose normalization, pose fitting and pose adaptive filtering), one common strategy is to perform face normalization to transform given test face to frontal pose. Pose normalization allows the direct use of conventional face recognition for matching. This section reviews several research methods that have been developed to perform the pose normalization. Pose normalization can be done in 2D domain or 3D domain [5], [11]-[12]. The 2D pose normalization-based methods conduct face synthesis either pixel wise, patch wise or region wise on the face image. Early 2D pose normalization works for face synthesis are based on forming the Delaunay triangles across the dense facial landmarks on the face. Delaunay triangulation was followed by the warping transformation which can be affine or thin plate splines between each pair of triangle regions of the gallery image and the probe image [14], [15]. In spite of simplicity, this approach writhes from stretching artifacts which degrades further with the degradation of image quality. Chai et al [9] conducted face normalization in a patch-wise strategy by performing linear mapping approximated by linear regression modeling between a set of non-frontal face patches and the corresponding frontal face patches. The key assumption of the linearity of the manifold structure of a local patch does not hold true always and hence suffers from over-fitting problem i.e. the learned pose manifold structure may not realistically represent the structure in another pose. Annan et al [16] improved the over-fitting issue by imposing ridge regularization constraint. Regularized learning is used to avoid over-fitting, where large features weights are penalised. Arianpour et al [17] proposed a nonlinear regression method (kernel-based method) and Hassner et al [23] proposed synthesizing the virtual frontal view from a non-frontal face input. In order to improve the pose-invariant face recognition rate, a nonlinear mapping kernel function  $\Psi$  was proposed with a purpose to map the input data space  $\Psi^n$  to the high dimensional feature space [9]. The choice of nonlinear mapping function  $\Psi$  was left as free parameter, though its function was defined to find the inner product in feature space between vector pairs. It is defined as:

$$\kappa(x_i, x_j) = \langle \Psi(x_i), \Psi(x_j) \rangle = \Psi(x_i)^T * \Psi(x_j) \quad (1)$$

In general, there are four important kernel functions [18] in pattern recognition and image processing applications, named linear, sigmoid, Gaussian and polynomial kernels. For any image vectors  $(x_i, x_j)$ , belonging to the input space, the linear kernel [18] is defined as:

$$\kappa(x_i, x_j) = x_i^T x_j + C \quad (2)$$

where C is optional constant parameter. Polynomial kernel [18] is defined as:

$$\kappa(x_i, x_j) = (\beta x_i^T x_j + c)^d \quad (3)$$

where c is constant, d is the degree of polynomial and  $\beta$  is adjustable parameter. In case where the training data has been normalized, the Polynomial kernel application is well suited. The Gaussian kernel [18] is defined by:

$$\kappa(x_i, x_j) = \exp(-(\|x_i - x_j\|)^2 / 2\sigma^2) \quad (4)$$

where  $\sigma$  is the standard deviation and its role in its application is very important. In case of overestimation, the exponential function will behave almost linear and also its higher-dimensional projection will miss its nonlinear power [21]. In case of underestimation, the regularization capability of the function will be decreased. Recently, instead of using one kernel, multiple kernel learning is seen as an alternative to overcome limitations of single kernel [20], [22]. Multiple kernel learning combines linear/ non-linear combination of predefined set of kernels. Simple linear combination of kernels combines kernels either by considering sum or mean of the kernels. Zhang et al. [22] proposed a valid ELM nonlinear mixed kernel function derived from the product of generalized Hermite Dirichlet kernel and triangular kernel.

$$K_{Tri-H}(x, z) = (1 - \|x - z\|/\sigma_0) * \sum_{i=1}^n H_i(x) H_i^T(z) \quad (5)$$

They proved their mixed kernel satisfy Mercer's condition and hence can be used a valid kernel function of extreme learning machine. Its applicability in the regression framework was proved by experimental results obtained from regression datasets of the UCI benchmark repository.

### B. Extreme Learning Machine

Extreme Learning machine(ELM)- A single hidden layer feed forward neural network, proposed by Huang *et al* [1],[2],[4] emerged as one of the appealing approach that has powerful learning abilities from a large amount of training data with fast speed and good generalization performance. Given a database set of N arbitrary distinct samples  $\{(x_i, t_i) \in R^d \times R^c\}$ , where  $x_i = [x_{i1} \dots x_{id}]^T \in R^d$  and  $T_i = [t_{i1}, t_{i2}, \dots, t_{ic}]^T \in R^c$ , d is number of features and C is number of classes. The objective is to find the weight vector  $\beta = [\beta_1, \beta_2, \dots, \beta_L]$  which will minimize  $\|H\beta - T\|^2$  and output weight  $\|\beta\|$ , where  $H$

$$H = [h(x_1), h(x_2), \dots, h(x_d)]' \quad (6)$$

is the hidden-layer output matrix

#### Standard three step ELM Algorithm for Classification

1. Assign randomly input weight vectors  $a_i$  and hidden node bias impact factor  $b_i$ ,  $i = 1 \dots L$ .  $a_i = [a_{i1}, a_{i2}, \dots, a_{id}]^T$  connects the  $i^{\text{th}}$  weights of hidden layer nodes to input.

2. The hidden layer output matrix H is calculated by the use of non-linear activation function

$$\begin{bmatrix} g(x_1 \cdot w'_1 + b_1) & \dots & g(x_L \cdot w'_L + b_L) \\ \vdots & & \vdots \\ g(x_N \cdot w'_1 + b_1) & \dots & g(x_L \cdot w'_L + b_L) \end{bmatrix} \quad (7)$$

In case where  $g(x)$  is unknown then kernels can be used in ELM [3].

3. Finally, the output weight  $\beta$  is calculated by using  $\beta = H^\dagger T$  where  $H^\dagger$  represents the hidden layer output as Moore-Penrose generalized inverse. A positive value is added to the diagonal of  $H^T H$  or  $H H^T$  in-order to make the resultant solution more stable and to have improved generalization performance i.e.

$$\beta = H^T (H H^T + I/C)^{-1} T \quad (8)$$

ELM framework can be represented in the equation:

$$g(x) = \sum_{i=0}^L \beta_i \cdot g(a_i \cdot x_j + b_i) = t_j$$

where  $j = 1, 2, \dots, N$  (number of samples taken).

Recently Goel *et al* [8] used kernel ELM regression framework to normalize the given input face poses to frontal pose. In kernel version of ELMs, only dot products of the data representations from the ELM space are used as the network outputs. However, for all the calculations they used maximal dimensionality of the training manifold.

### III. PROBLEM FORMULATION AND PROPOSED APPROACH

Pose variation introduces high variance in the coefficients estimated from 2D images of the same subject. The representation accuracy of features degrades with the introduction of high variance. Due to pose variation reliable and discriminative features sometimes also undergoes self-occlusion. Problem of face classification across the poses naturally transforms into regression problem. Accurate estimation of non-linear changes cannot be estimated by linear regression. In this paper non-linear regression of images with reduced coefficients via kernel ELM is proposed to estimate the non-linear mapping between frontal face views from its counter-part non-frontal views for effective face recognition.

Data set is divided into training set and testing set. In training set as well as testing set, there exist two matrices  $F = [f_1, f_2, \dots, f_n] \in R^{D \times N}$  and  $P = [xp_1, xp_2, \dots, xp_n]$  where  $f_i, i=1 \dots N$  a set of N front face,  $xp_i, i=1 \dots N$  a set of N pose faces and the corresponding class labels  $li \in \{1, \dots, C\}$ .  $K$  is kernel function defined on Train set images and  $K'$  is kernel function defined on Train and Test set images  $Pp$  that can be used to define training kernel set.

**Input:** Set of Pose Images  $P_p = [xp_1, xp_2, \dots, xp_n] \in R^{D \times N}$

**Output:** Front pose  $K * P_p = F_0 \in R^{D \times N}$  (9)

Here  $K$ , denotes the output weights of the hidden layer. The architecture of Kernel ELM for single hidden layer feed

forward network, for the given training set consist of  $d$  units from pose face image set for input and  $d$  units output layer comprising of frontal face image, and kernel matrix  $K_{N \times N}$  acting hidden layer neurons.

Algorithm for pose normalization using kernel-ELM based nonlinear regression with reduced coefficients

1. Given a training set containing N samples of different poses including frontal ones  $F_{Train}(F_{Tri})$ ,  $PoseTrain(P_{Tri})$   $i=1, 2, 3, \dots, N$  with  $m$  classes.
2. All the N samples are pre-processed with Gamma correction function which replaces gray-levels in the image with new gray levels ( $I^\alpha$ ), where  $\alpha$  is a user-defined parameter. This step enhances the local dynamic range of the image in dark or shadowed regions while compressing it in bright regions.
3. After gamma processing each of the sample image space is transformed to transformed space by applying DCT (Discrete Cosine Transformation) blockwise with block size of 8x8 blocks.
4. From 64 coefficients of each block 25 low frequency coefficients are retained.
5. Image space with reduced coefficients is retrieved back by using Inverse DCT.
6. Train kernel and Test kernel matrix is calculated from N samples with reduced coefficients instead of activation function  $g(x)$  for hidden nodes [3]. The kernel matrix:

$$\Omega_{ELM} = \Omega_{ELM_{i,j}} = h(\mathbf{x}_i) \cdot h(\mathbf{x}_j) = K(\mathbf{x}_i, \mathbf{x}_j) \quad (10)$$

is only related to the input data  $\mathbf{x}_i$  and the number of input samples.

7. Output weight is computed by  $\mathbf{f}(\mathbf{x})$  which can now be expressed in terms of

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} K(x, x_1) \\ \vdots \\ K(x, x_N) \end{bmatrix}^T (\lambda I/C + \Omega_{ELM})^{-1} T \quad (11)$$

where  $T$  is the front pose. Parameter  $\lambda$  controls the amount of shrinkage of coefficients.

8. Compute Testing and Training Accuracy

### IV. EXPERIMENTAL RESULTS

In all the experiments, FERET [19] database with pose variations has been used to show the performance of the KELM regression with low rank decomposition. KELM regression with low rank decomposition is used to normalize the pose variations for face recognition. Database subset used for this paper contained total 200 subjects including 7 pictures per subject with different pose views at pose angles  $0^\circ, \pm 10^\circ, \pm 20^\circ, \pm 30^\circ$  and  $\pm 45^\circ$ . In this study, each subject with 5 poses at angles  $0^\circ, +10^\circ, +20^\circ, +30^\circ$  and  $+45^\circ$  is considered. The resolution of each cropped image is 80x80 and is in '.tif' format. Five images of each person were randomly taken as the training sample, the other two frontal and near frontal images of each person is taken as the test set. For experimental results, programs were designed using

MATLAB 2014b on a computer with 8 GB RAM and core i3 3.00 GHz processor.



Fig. 1. Virtual View Generated by Linear Regression

Fig. 1 shows the front view synthesized by using Linear Regression. This method though easy to implement shows ghost effects in the virtual front pose. Fig. 2 represents virtual view generated by Kernel version of ELM regression with reduced image space. Linear kernel is used since the number of features per image is very large. Regression coefficient is added to improve the KELM regression performance. Other kernels like RBF kernel, exponential chi square kernel generate sparse outputs with reduced image space and hence they are not suitable for kernel ELM regression. Detailed texture details are lost in the virtual view generation with reduced coefficient set though the virtual view can be correlated with the actual front view with good correlation factor.

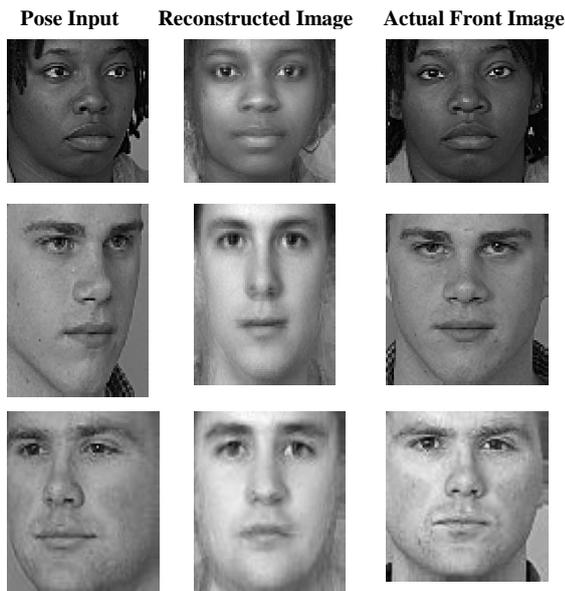


Fig. 2. Virtual View Generated by KELM regression with reduced image coefficients

Fig. 3 depicts the virtual view generation with KELM regression without reduced coefficients and without any gamma processing of images. Certain artifacts are also introduced in the front virtual views.

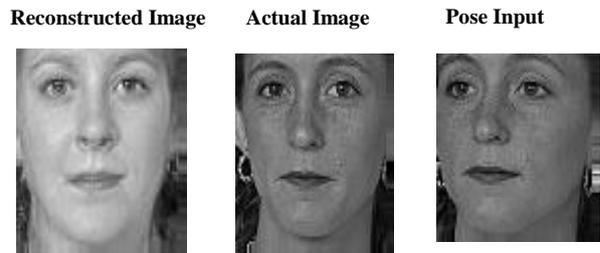


Fig. 3. Virtual Generated by KELM Regression with out pre-processing

Cross Correlation coefficient, Structural Similarity Index (SSIM) value and mean squared error are used to evaluate the similarity between the generated front image and the actual front pose. Table I illustrates the similarity between input set and the generated image set.

TABLE I: SIMILARITY INDEXES BETWEEN VIRTUAL AND ACTUAL FRONTAL IMAGES

Evaluation Criteria	Cross Correlation	SSIM(Structural Similarity Index)	MSE(mean squared error)
Linear Regression	.899	.887	202.239
KELM Regression	1.121	0.934	304.458
KELM Regression With reduced Coefficients	.896	0.908	214.356

We also studied comparative performance for various kernel functions in terms of accuracy for test sets for pose at 45 degrees and the time required for training and testing for given kernel functions. Results are presented in Table II. For Testing accuracy, Structural Similarity Index (SSIM) value is used to evaluate the similarity between the generated front image and the actual front pose.

TABLE II: COMPARATIVE ANALYSIS FOR DIFFERENT KERNEL FUNCTIONS FOR KELM REGRESSION

Kernel Function	Testing accuracy( %)	Training Time (sec)	Testing Time (sec)
Linear	89	.0167	.0203
Polynomial	90.67	.0198	.0234
Fractional power polynomial (d=1.85)	91.09	.0269	.0341
Gaussian	79.168	0.838	0.892
Triangular Kernel	76.23	.034	.0353
Laplacian Kernel	76.956	.612	.661
Mixed Kernel $K_{Tri-H}(x,z)$	92.02	.025	.0268

All the kernel functions (radial basis ,laplace and traingular) dependent upon sigma parameter  $\sigma_0$  requires data set appropriate sigma value. Small value of sigma may generate good accuracy for training data set but not for test data set. A very large value of sigma for Gaussian kernel of RBF makes it to behave almost like linear function. For the results mentioned in the paper  $\sigma_0$  is taken as twice the maximum distance of all sample points to sample mean such that formula holds with probability one [22]. Fig. 4 shows results with sigmoid kernel function. Data over fitting is very high with sigmoidal kernel function resulting in inappropriate calculation of regression coefficients for test set.

Reconstructed Image Actual Image Pose Input



Fig. 4. Virtual View Generated by KELM Regression with sigmoid kernel function and inappropriate sigma value

Fig. 5,6,7 present results with mixed kernel obtained by product of generalized Hermite Dirichlet kernel and triangular kernel function. Table III lists the Generalized Triangular Hermite kernel function upto 2<sup>nd</sup> order.

TABLE III: LIST OF GENERALIZED TRIANGULAR HERMITE KERNEL FUNCTION UP TO SECOND ORDER [22]

Order (n)	$K_{Tri-H}(x,z)$
0	$1 \cdot (1 -  x - z /\sigma_0)$
1	$(1 + 4xz^T) \cdot (1 -  x - z /\sigma_0)$
2	$[1 + 4(2\ x\ ^2 - 1) \cdot (2\ z\ ^2 - 1) + 4xz^T] \cdot (1 -  x - z /\sigma_0)$

Tri-H KELM regression for order 2 and order 1 is able to obtain almost similar regression coefficients and is able to retain discriminative information among various subjects. Order 0 is although able to retain information at contour portion but not at the surface where the intensity change is invariably not much. However, Tri-H KELM for face normalization is not able to retain texture information. Underlying reason for poor performance can be accounted for inability to retain similarity information of testing and training data.



Fig.5. Virtual View Generated by Mixed KELM Regression level 2



Fig.6. Virtual View Generated by Mixed KELM Regression level 1



Fig.7. Virtual View Generated by Mixed KELM Regression level 0

We also evaluated various features with ELM classification framework. For experiments we have 1500 hidden neurons. Table IV indicates the ELM classification accuracy with features that can tolerate pose variations up to certain extent. Standard ELM algorithm [2] is used to evaluate the features.

TABLE IV: TRAIN AND TESTING ACCURACIES

Feature Set	Training	Testing
	Accuracy (%)	Accuracy(%)
<b>Entropy and intensity values</b>	100	91.10%
<b>Uniform LBP(Linear Binary Pattern)</b>	96.54	82.46
<b>Entropy features</b>	89.9	78.25
<b>HOG (Histogram Oriented Features)</b>	94.23	87.8
<b>Pixel /Intensity Values</b>	89	74

Among the various feature evaluated from the posed inputs we found the combination of intensity and entropy values works best. The combination of appearance based and transformation based features works better than individual feature sets.

## V. CONCLUSION

With KELM regression framework, performance of pose normalization with reduced coefficients is found to be comparable to KELM regression. However as the pose difference increases from the frontal face the synthesized face image is found to contain significant stretching artifacts. Pose synthesized by KELM is able to retain facial contour information of the original 2D images which is an important parameter for establishing identity but the regressed pose is not good in dealing with nonlinear texture warping. The proposed approach is able to synthesize the input pose with much reduced computations as compared to its counterpart's approaches that use 3D approaches or combination of 2D and 3D pose synthesis approach. This paper also explored the practical application of mixed kernels (Tri-H KELM) for finding regression coefficients from posed input face data to map it to virtual front pose. It was found that mixed kernel(Tri-H KELM) is able to achieve better generalization performance than the polynomial as well as sigma dependent kernel (RBF, Laplace, Triangular) functions when the sigma value is inappropriately chosen. With ELM classification framework, good classification accuracy is obtained when combination of intensity and entropy based features is used. However the empirical tuning of hidden neurons remains a tricky issue here. For future work, we will try to explore more complimentary global and local features robust to poses. We will also try to explore the impact of imposing cascade of regularization constraints.

## REFERENCES

- [1] G.B. Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: a new learning scheme of feedforward neural networks", *Proc.IEEE International Joint Conference on Neural Networks*, 2, pp. 985-990,2004.
- [2] G. B. Huang, Q.Y. Zhu, and C.K. Siew., "Extreme learning machine: theory and applications", *Neurocomputing*, 70(1), pp.489-501,2006.
- [3] G.B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 42(2), pp.513-529,2012.
- [4] G. B. Huang, "An insight into extreme learning machines: random neurons, random features and kernels," *Cognitive Computation*, 6 (3), pp.376-390,2014.
- [5] C. Ding and D. Tao, "A comprehensive survey on pose-invariant face recognition," *ACM Transactions on intelligent systems and technology (TIST)*, 7(3), p.37,2016.
- [6] Z. Bai, G. B. Huang, , D. Wang, H. Wang, and M. B. Westover, "Sparse extreme learning machine for classification," *IEEE transactions on cybernetics*, 44(10), pp.1858-1870,2014.
- [7] A. Iosifidis, A. Tefas, and I. Pitas, "On the kernel extreme learning machine classifier," *Pattern Recognition Letters*, 54, pp.11-17,2015.
- [8] T. Goel, V. Nehra, and V. P. Vishwakarma, "Pose Normalization based on Kernel ELM Regression for Face Recognition", *International Journal of Image, Graphics & Signal Processing*, 9 (5),2017.
- [9] X. Chai, S. Shan, and W. Gao, "Locally linear regression for pose-invariant face recognition," *IEEE Transaction on Image Processing*, 16(7), pp. 1716–1725,2007.
- [10] C.K. Hsieh, and Y.C. Chen, "Kernel-based pose invariant face recognition," *In Multimedia and Expo, 2007 IEEE International Conference*, pp: 987-990,2007.
- [11] D. Yi, Z. Lei, and S. Z. Li, "Towards pose robust face recognition," *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3539-3545,2013.
- [12] A. Asthana, T.K.Marks, M.J.Jones, K.H.Tieu, and M.V. Rohith, "Fully automatic pose-invariant face recognition via 3D pose normalization," *In Computer Vision (ICCV), IEEE International Conference*, pp. 937-944,2011..
- [13] Phillips, P.J., The facial recognition technology (FERET) database. (2004)
- [14] D.Reisfeld, Y.Yeshurun, "Preprocessing of face images: Detection of features and pose normalization," *Computer vision and image understanding- Elsevier*,1998.
- [15] H.Gao , H.K.Ekenel, and R.Stiefelhagen , "Pose normalization for local appearance-based face recognition," *In Proc. Int. Conf. Adv. Biometrics*, pp. 32–41,2009.
- [16] L.Annan., S.Shan , and W.Gao , "Coupled bias–variance tradeoff for cross-pose face recognition," *IEEE Trans. Image Processing*., 21(1),pp. 305–315,2012.
- [17] Y.Arianpour, S. Ghofrani,& H.Amindavar, "Locally Kernel-based Nonlinear Regression for Face Recognition," *International Journal of Signal Processing, Image Processing and Pattern Recognition*, 5(4), pp. 131-146,2012.
- [18] B.Scholkopf and A.Smola, "Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond," Cambridge, MA: MIT, 2002.
- [19] A.E. Hoerl and R.W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, 12(1), pp. 55–67,1970.
- [20] Z. Xie , P. Jiang & S.Zhang, "Fusion of LBP and HOG using multiple kernel learning for infrared face recognition," *In Computer and Information Science (ICIS), IEEE/ACIS 16th International Conference*, pp.81-84, 2017.
- [21] M. Tian and W. Wenjian, "Some sets of orthogonal polynomial kernel functions," *Applied Soft Computing*, vol. 61, pp. 742-756, 2017.
- [22] S. Zhang and T. Wenan, "An Extreme Learning Machine Based on the Mixed Kernel Function of Triangular Kernel and Generalized Hermite Dirichlet Kernel," *Discrete Dynamics in Nature and Society*, 2016.
- [23] T.Hassner, S.Harel , S.Paz, and R.Enbar , "Effective Face Frontalization in Unconstrained Images," *Proc. IEEE Conf. Comput. Vis. Pattern Recognition*,pp. 4295–4304, ,2015.