Convolutional Neural Network for Face Recognition with Pose and Illumination Variation

A. R. Syafeeza [#1], M. Khalil-Hani [#2], S. S. Liew [#3], R. Bakhteri [#4]

VeCAD Research Laboratory,
Faculty of Electrical Engineering, Universiti Teknologi Malaysia,
81310 Skudai, Johor, Malaysia
[1] syafeeza@utem.edu.my
[2] khalil@fke.utm.my
[3] gladion89@live.com
[4] rabia@fke.utm.my

Abstract— Face recognition remains a challenging problem till today. The main challenge is how to improve the recognition performance when affected by the variability of non-linear effects that include illumination variances, poses, facial expressions, occlusions, etc. In this paper, a robust 4-layer Convolutional Neural Network (CNN) architecture is proposed for the face recognition problem, with a solution that is capable of handling facial images that contain occlusions, poses, facial expressions and varying illumination. Experimental results show that the proposed CNN solution outperforms existing works, achieving 99.5% recognition accuracy on AR database. The test on the 35-subjects of FERET database achieves an accuracy of 85.13%, which is in the similar range of performance as the best result of previous works. More significantly, our proposed system completes the facial recognition process in less than 0.01 seconds.

Keyword- Convolutional Neural Network, face recognition, biometric identification, Stochastic Diagonal Levenberg-Marquardt

I. INTRODUCTION

A robust human identity authentication system is vital nowadays due to the increasing number of crime and losses through identity fraud. As the traditional token-based and knowledge-based system possess high risks of being stolen or forgotten, biometric systems are applied in wide range of applications such as access control systems, criminal identification, autonomous vending and automated banking due to the strengths of the unique biometric feature and non-transferable characteristic [1]. Biometrics is divided into two categories of either behavioural (typing rhythm, gait or voice) or physiological (fingerprint, vascular, face, iris or signature). Among the aforementioned of physiological biometric identifiers, face recognition system remains as a valid research area since 1960s with wide room for improvements.

There are several factors why face recognition is a challenging problem. One factor is due to varying poses of the facial image. The relative camera may either capture frontal, 45°, profile or upside down facial images. Certain poses cause some of the facial features such as eyes or nose to be partially occluded. Another factor is due to presence or absence of structural components such as beard, moustaches, with/without glasses in the image. These structural components have a great deal of variability including shape, colour and size. Other factors that can affect accuracy include illumination, occlusion and facial expressions. Illumination is a change of light ambient due to skin reflectance properties and internal camera control that may cast shadow on some part of the face. Occlusion is the result of an object covering part of the face, such as a scarf, turban, etc. Examples of facial expressions are smile, laugh, anger, sad, surprise, disgust, and fright [1].

Biometric recognition system generally falls into two categories: verification and identification. Biometric verification is a method that performs one-to-one comparison between the query template of a subject with the template of the claimed identity stored in database. On the other hand, biometric identification performs one-to-many matches between an unknown subject and the references or templates stored in the database (biometric templates in databases are often divided into categories based on the biometric pattern or soft-biometrics for fast one-to-many identification process). The system returns the identity of the unknown subject after the matching process. Between these two modes, identification is more challenging as the matching process is time consuming and the chances of having false acceptance is much higher [1].

A well-known candidate in handling the stated problems is neural network. Neural network is robust in handling noisy image and has the ability to learn from experience. It is formed by several processing elements that provide the ability to parallelize such processing. Inspired by the human brain, a neural network or typically known as multilayer perceptron (MLP) works as a powerful classifier. However, MLP requires the input image

to undergo several image processing tasks such as pre-processing, segmentation and feature extraction in order to provide good performance. MLP suffers from the existence of free parameters (redundant information) in its architecture. These free parameters are formed by full connection scheme between the input and the feature maps from the following layer.

A variant of MLP that overcomes the constraints mentioned above is known as Convolutional Neural Network (CNN). CNN is inspired by visual mammalian cortex of simple and complex cells [2]. It consists of 4-8 layers with image processing tasks incorporated into the design. CNN applies three architectural concepts in its architecture namely shared weights, local receptive field and subsampling. Shared weights are formed by convolution kernel in order to reduce a number of free parameters. These convolution kernels are randomly initialized and form a noise filter as well as edge detector for the feature maps at the respective layer. Local receptive field is applied by both convolution and subsampling kernel by taking a group of neighbourhood pixels for further processing before passing the result of a coarser resolution to the following CNN layers. Subsampling process forms local averaging while reduces the feature map size at the respective layer. At this point, the exact locations of the features are not important anymore. This process adds the robustness of CNN towards handling deformation of input images such as translation, rotation and scaling [2].

CNN is unique due to the architecture itself. It performs segmentation, feature extraction and classification in one processing module with minimal pre-processing tasks on the input image. Minimal domain knowledge of the problem at hand is sufficient to perform efficient pattern recognition tasks. This has been proven by a wide range of applications that are using CNN such as face detection [3], face recognition [4], gender recognition [5], object recognition [6], character recognition [7], texture recognition [8] and so forth. This is completely in contrast with the conventional pattern recognition tasks in which prior knowledge of the problem at hand is needed in order to apply a suitable algorithm to extract the right features. In addition, whenever the problem domain changes the whole system needs to change requiring a re-design of the algorithm from the start [2].

In this paper, we propose two 4-layer CNN architectures for face recognition system. The first CNN architecture is designed for frontal images with occlusion, illumination variances and facial expressions. The second CNN architecture is designed for various poses, illumination variances and facial expressions. The proposed architecture is much simpler than the common LeNet-5 architecture but still maintains the efficiency in generalizing unseen data. It also recognizes a face within less than 0.01 seconds.

The remainder of this paper is organized as follows. Section II discusses on the related work. This is followed by the proposed network architecture in section III. Experimental results and benchmarking of results with existing works are discussed in section IV. Ultimately, the final section concludes the work.

II. Related Work

There are three general approaches for existing face recognition works namely appearance-based (or holistic matching methods), feature-based and hybrid approach. These approaches are distinguished based on the feature extraction method.

In appearance-based method, the whole face region is accepted as the raw input to a recognition system. The facial input image is transformed into a face space analysis problem and followed by a number of well-known statistical methods. Methods that lie under this category are eigenface [9], fisherface [10], frequency domain [11], independent component analysis (ICA) [12], support vector machines [13], Laplacian [14] and probabilistic decision based neural network [15] face approach. Linear dimensionality reduction technique is applied in this category which makes recognition process under illumination changes often fail.

Feature-based method extracts local feature information such as mouth, eyes and nose. The location and local statistics (geometric and/or appearance) of these features are then fed into a structural classifier. Examples of the techniques under this category are geometrical feature [16], EBGM [17], HMM [18], active appearance model-2D [18] and 3D morphable model [19]. Generally, the methods under this category require empirical choice of parameters such as number of scales and orientations of the filters. On top of that, the designers have to specify appropriate area to apply the filters which makes their implementation sub-optimal.

Hybrid method is the third approach which combines the appearance-based and feature-based methods to recognize faces. Examples under this category are modular eigenfaces and eigenmodules [20], hybrid local feature analysis [21], shape-normalized flexible appearance models [22] and component-based face region and components [22]. The methods under this category increase the design complexity by merging two methods and some of the hybrid techniques produce lower accuracy compared to the individual method.

CNN falls under the second category but the position and values of filters are automatically decided by the CNN during training process. LeNet-5 is the classical CNN proposed by LeCun et al. in 1998, which was applied in handwritten digit character recognition [2]. There have been several works that use CNN to tackle face recognition problem. Among the early works on face recognition research using CNN was reported by Lawrence et al. in 1997 [23]. In their work, input samples were first processed by a Kohonen self-organizing map neural network to reduce dimensionality, and then followed by the operation on a LeNet-5 CNN. Learning

was performed using a standard backpropagation algorithm. Their method had high complexity since two different neural networks were combined to perform the recognition tasks. In 2007, Duffner and Garcia, displayed a unique approach compared to other CNN-based work [24]. They trained the system to transform an input image into a reference image predefined for each subject. More recently, Khalajzadeh et al. [25] proposed a hierarchical structure based CNN which was tested on ORL, Yale and JAFFE databases. It has 4 layers and standard backpropagation is applied as the learning algorithm. Both of these approaches have achieved unsatisfying recognition rate.

III. PROPOSED NETWORK ARCHITECTURE

The common LeNet-5 architecture has a convolution and subsampling layer that are alternated twice. In this work, we fuse the convolution and subsampling layer and form a simplified version of CNN. We adapt the idea from the work of Y.Simard et al. in which they applied this fusion approach to handwriting recognition problem with 10 classes [26]. Mamalet and Garcia [27] confirmed the viability of this approach in accelerating the processing time. The idea of this fusion approach is adapted to face recognition problem for 100 classes and incorporate partial connection between the first two layers to ensure different features are learned during the training process. Comparison between the common CNN approach and fusion approach is depicted in Fig. 1. It can be observed that the common approach requires two stages to perform convolution and subsampling whereas the fusion approach requires only one stage. Padding is not required during the convolution process and therefore reduces the size of feature maps at the succeeding layer.

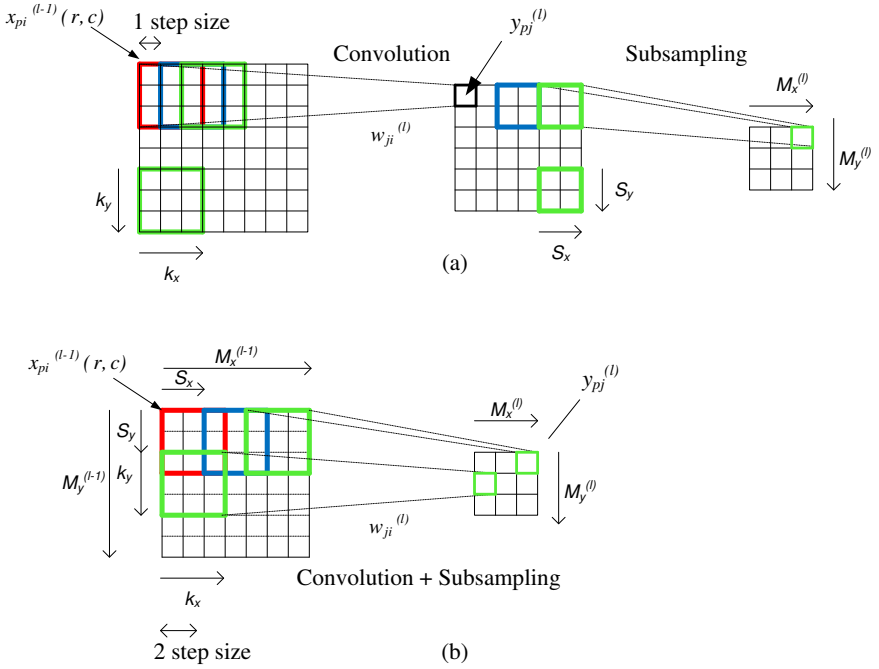

Fig. 1. Comparison between (a) common convolution and subsampling method in LeNet-5 architecture and (b) fusion approach

In the fused convolution/subsampling process, the convolution kernel $w_{ji}^{(l)}$ is convolved with the input feature map M_i, with the subsampling incorporated as a skipping operation in the convolution process. The average pooling operation in a subsampling layer is not performed. Instead, only the sizes of the kernels (S_x, S_y) are represented as skipping factors between subsequent convolutions in x and y directions respectively. Equation (1) gives the sizes of the output map (M_x, M_y), in x and y direction.

$$M_x^{(l)} = \frac{M_x^{(l-1)} - \left(K_x^{(l)} - S_x^{(l)}\right)}{S_x^{(l)}}; M_y^{(l)} = \frac{M_y^{(l-1)} - \left(K_y^{(l)} - S_y^{(l)}\right)}{S_y^{(l)}} \tag{1}$$

where (K_x, K_y) denotes the convolution kernels, index (l) indicates the layer. Equation (2) is the mathematical form of the output unit of a convolutional layer:

$$y_{pj}^{(l)} = f\left(\sum_{i \in M_j^{(l-1)}} \sum_{(u,v) \in K^{(l)}} w_{ji(u,v)}^{(l)} * x_{pi}^{(l-1)}(c+u, r+v) + b_j^{(l)}\right) \tag{2}$$

where $K = \{(u,v) \in N^2 \mid 0 \le u < k_x; 0 \le v < k_y\}$, k_x and k_y are the width and the height of the convolution kernels $w_{ji}^{(l)}$ of layer *(l)*, $b_j^{(l)}$ is the bias of feature map *n* in layer*(l)*, *c* and *r* refers to the current pixel and *p* refers to the particular training sample. The set $M_j^{(l-1)}$ contains the feature maps in the preceding layer *(l-1)* that are connected to feature map *n* in layer *(l)*. The notation *f* is the activation function of layer *(l)*. The variable *u* and *v* describes the horizontal and vertical step size of the convolution kernel in layer *(l)*.

The connection between layer C3 and output is a full-connection layer that is basically a MLP. Equation (3) gives the mathematical form of full-connection layer:

$$y_{pj}^{(l)} = f\left(\sum_{i=0}^{N^{(l-1)}} x_{pi}^{(l-1)} \cdot w_{ji}^{(l)} + b_j^{(l)} \right) \tag{3}$$

where $N^{(l-1)}$ is the number of neurons in the preceding layer *(l-1)*, $w_{ji}^{(l)}$ is the weight for connection from neuron *i* in layer *(l-1)* to neuron j in layer *(l)* and $b_j^{(l)}$) is the bias of neuron *j* in layer *(l)*, and *f* represents the activation function of layer *(l)*.

The CNN architecture for AR database and FERET database are depicted in Fig. 2 and Fig. 3 respectively. The AR and FERET database require different CNN architectures for optimal performance. This is because the two databases differ in terms of data complexity. For instance, AR consists of frontal images with occlusion, illumination variances and facial expressions while FERET consists of various poses, illumination variances and facial expressions. Information for each database is discussed in details in Section IV.

Via a 10-fold cross-validation model selection method, the architecture for AR and FERET database has been experimentally chosen from a selection of six different models to have the lowest validation error. The parameters governing the six different CNN models include the number of layers; number of feature map in each layer and their connectivity. Generally, 20% of the total samples are taken to form the test dataset. As we are performing 10-fold cross-validation to find the best model complexity, the remaining 80% samples are divided into two datasets: a validation set and a training set. These images are first divided into 10 folds, of which one is selected to be the validation dataset, and the remaining 9 folds are combined to form the training dataset. This method is repeated ten times, each time selecting a different fold as a validation set and the corresponding remaining nine to form the training set. These validation and training datasets are used in the 10-fold cross-validation applied in the experiment to avoid over-fitting and find the best model complexity. In order to evaluate the actual performance, test samples are used to measure the generalization ability of the network. In this work, full-connection scheme is deployed during the best model selection process, and Stochastic Diagonal Levenberg Marquardt (SDLM) is the learning algorithm applied. It has a regularization parameter to be tuned. Details of this algorithm can be referred to [2, 28].

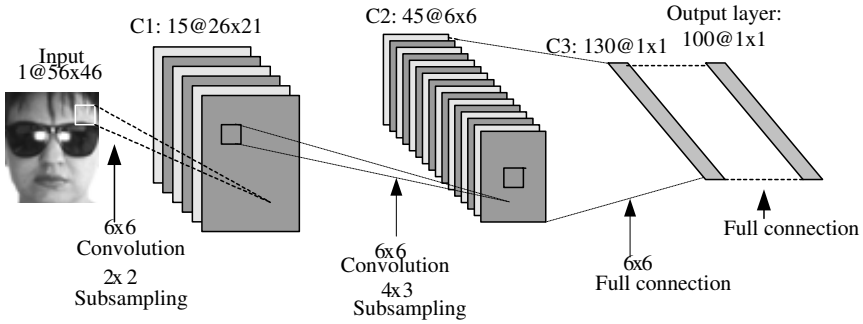

Fig. 2. The proposed architecture (referred here as the 15-45-130 model) for AR database

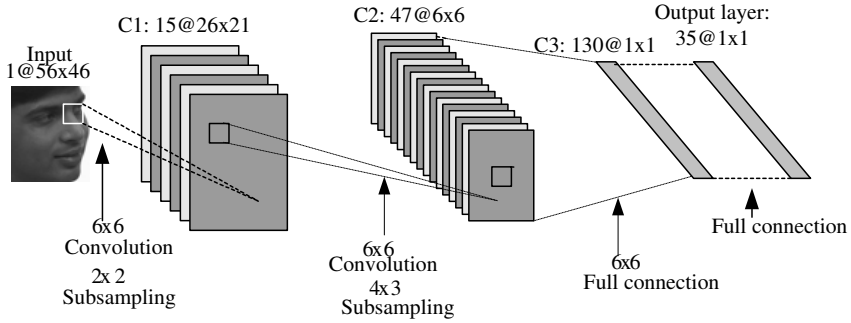

Fig. 3. The proposed architecture (referred here as the 15-47-130 model) for FERET database

We apply partial connection between layer C1 and C2 to break connections. The connection map is decided after implementing 10-fold cross-validation approach. This approach can also reduce significant number of trainable parameters to learn (trainable parameters impact on memory consumption). The purpose of conducting this process is to make sure that each feature map at layer C2 learns different features and avoid data redundancy. Various types of connections are shown in TABLE I. The notation "×" indicates connection of feature maps at layer C1 and C2 respectively. Before conducting this process, full-connection scheme is assigned as the default connection map between these layers. By referring to TABLE II and TABLE III, each column is evaluated with eight types of connection maps (as shown in TABLE I) starting from the first column, C2(0). The map that produces the lowest validation error is selected. Once the best connection map is identified for C2(0), this connection is fixed and the same process is continued until the second last column (C2(44) for AR database and C2(46) for FERET database). Only the last feature map from layer C2 receive all input feature maps from layer C1. An optimized version of partial connection for AR and FERET database are depicted in Table II and Table III respectively.

TABLE I
Variant type of connections between C1 and C2 layer

conn(0)	conn(1)	conn(2)	conn(3)	conn(4)	conn(5)	conn(6)	conn(7)
×							
×	×						
×	×	×					
×	×	×	×				
×	×	×	×	×			
×	×	×	×	×	×		
×	×	×	×	×	×	×	
×	×	×	×	×	×	×	×
	×	×	×	×	×	×	×
		×	×	×	×	×	×
			×	×	×	×	×
				×	×	×	×
					×	×	×
						×	×
						×	×
							×

TABLE II
Connection map for AR database

C1 \ C2	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44
0								x								x								x								x												x
1					x			x						x		x				x				x								x					x			x				x
2					x	x		x			x			x		x				x				x						x		x				x			x				x
3					x	x		x		x	x			x		x				x				x			x			x		x			x		x	x				x
4					x	x		x	x	x	x		x	x	x	x	x			x		x	x	x		x		x			x	x		x			x	x			x	x
5					x	x		x	x	x	x		x	x	x	x	x			x	x	x	x	x		x	x	x	x	x	x		x		x	x	x	x		x	x		x	x
6		x	x	x	x	x	x	x	x	x	x		x	x	x	x	x	x		x	x	x	x	x		x	x	x	x	x	x	x	x		x	x	x	x	x	x	x		x	x	x
7	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
8	x	x	x	x	x	x	x		x	x	x	x	x	x	x		x	x	x	x	x	x	x		x	x	x	x	x	x	x	x		x	x	x	x	x	x	x	x	x	x	x	x
9	x	x	x	x		x	x		x	x	x	x	x		x		x	x	x		x	x	x		x	x	x	x	x	x	x	x		x	x	x		x	x		x	x	x	x	x
10	x	x	x	x			x		x	x		x	x		x		x	x	x		x	x	x		x	x	x	x	x		x	x		x	x	x		x	x		x	x	x	x	x
11	x	x	x	x			x		x			x	x		x		x	x	x		x	x	x		x	x	x		x		x	x		x	x			x	x			x	x	x	x
12	x	x	x	x			x					x						x	x		x				x		x		x			x		x	x			x	x			x	x		x
13	x	x	x	x			x					x						x	x						x							x		x				x				x	x		x
14	x											x							x						x							x										x			x

TABLE III
Connection map for FERET database

C1 \ C2	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46
0															x				x				x				x																				x	x
1						x	x			x					x				x	x			x				x																		x	x
2		x			x	x	x			x					x				x	x			x				x							x										x	x	x
3		x		x	x	x	x			x				x	x				x	x			x				x							x							x				x	x	x
4		x	x	x	x	x	x	x		x		x		x	x			x	x	x			x	x	x		x							x	x					x	x	x		x	x	x
5		x	x	x	x	x	x	x		x		x		x	x			x	x	x			x	x	x		x							x	x					x	x	x		x	x	x
6	x	x	x	x	x	x	x	x		x		x		x	x		x	x	x	x	x	x	x	x	x		x		x	x				x	x	x	x			x	x	x		x	x	x
7	x	x	x	x	x	x	x	x	x	x	x	x	x	x			x	x		x	x		x		x				x	x	x		x	x	x	x	x	x	x	x	x		x		x
8	x	x	x	x	x			x	x		x	x	x	x		x	x	x			x	x		x	x	x	x		x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		x
9	x		x	x				x	x		x	x	x	x		x	x	x			x	x		x	x	x	x		x	x	x	x	x		x	x	x	x	x	x	x	x	x			x
10	x		x					x	x		x	x	x			x	x	x			x	x		x	x	x	x		x	x	x	x	x		x	x	x	x	x		x	x	x	x		x
11	x								x		x		x			x					x	x				x	x		x	x	x	x	x			x	x	x				x	x	x		x
12	x								x		x		x			x	x				x	x				x	x		x	x	x	x	x			x	x	x				x		x		x
13									x		x		x			x										x	x		x			x	x					x				x		x		x
14																x										x	x					x										x		x		x

Hyperbolic tangent is used as the activation function in our design, as provided in equation (3). The notation x is the net input of a neuron. The activation function scales the net input into a specified range. In equation (4), the active range is [-1,+1] while the constant value is the asymptote of a sigmoid.

$$f(x) = 1.7159 \tanh\left(\tfrac{2}{3} x\right) \tag{4}$$

In this work, classification is based on winner-takes-all rule [29]. Neurons with the value of "+1" correspond to the "winning" neuron (that corresponds to the correct class) while "-1" corresponds to the incorrectly classified neurons. In winner-takes-all mechanism, each output neuron represents a single predefined class of the pattern set. It forces the output neurons to compete among each other, in which only the neuron with maximum output value is determined as the winner, which indicates the class that the particular input pattern belongs to. This approach is far simpler and computationally effective, since only the operation of finding maximum output value among these output neurons is required in order to determine the class of the pattern. By applying this approach, a true match is assigned during training and the weights are adapted according to the class assigned. This approach has overcome the limitation encountered by the conventional biometric approach in which after conducting a one-to-many matching between the unknown template and the ones stored in the database, it is not guaranteed that the true match is returned by the system. Instead, a list of candidates is returned. Hence, the conventional method is time consuming and less efficient.

The error function applied in this work is mean square error, E that is given by:

$$E(k) = \frac{1}{P \times M} \sum_{j=1}^{P} \sum_{i=1}^{M} \left(y_i^{\,j} - t_i^{\,j}\right)^2, k = 1,..., s_k \tag{5}$$

where, index P refers to the number of training patterns, M is the number of output neurons, s_k is the number of training iterations and, y_i^j and t_i^j is the actual and desired response of the output neuron due to the j^{th} input pattern.

IV. EXPERIMENTAL WORK AND RESULTS

A. Database and Data Preparation

The proposed design system ran on a 2.5 GHz Intel i5-3210M quad core processor, 8GB RAM, with Ubuntu Linux. MATLAB is utilized to prepare the samples in preprocessing steps (as described in the next section) and the algorithms are implemented in C. The proposed CNN architectures are tested on AR and FERET databases.

AR database is created by Aleix Martinez and Robert Benavente in the Computer Vision Center (CVC). It contains 3276 colour images corresponding to 126 subjects (70 men and 56 women) with 26 samples for each subject. It contains frontal view faces with high degree of variations in facial expressions, illumination conditions and occlusions (sun glasses and scarf). There were no restrictions imposed on participants especially on apparel, make-up, hair style, etc. Each subject participated in two sessions, separated by two weeks (14 days) time. The same pictures were taken in both sessions. The files are in BMP format, each of the images is 165×120 pixels in size. Examples of the image samples prepared for the following experiments are shown in Fig. 4. From the figure, images (a)-(m) and (n)-(z) are drawn from first and second session respectively. The first four images from each session i.e (a)-(d) and (n)-(q) involve variation of facial expression (neutral, smile, anger and scream). The next three images from each session i.e (e)-(g) and (r)-(t) involve variation of illumination condition (left light on, right light on and both sides light on).The images (h)-(j) and (u)-(w) are images with sun glasses and the remaining images from both sessions i.e (k)-(m) and (x)-(z) are images with partial occlusion (scarf covering mouth). In this work, we used 100 subjects only out of 126 subjects.

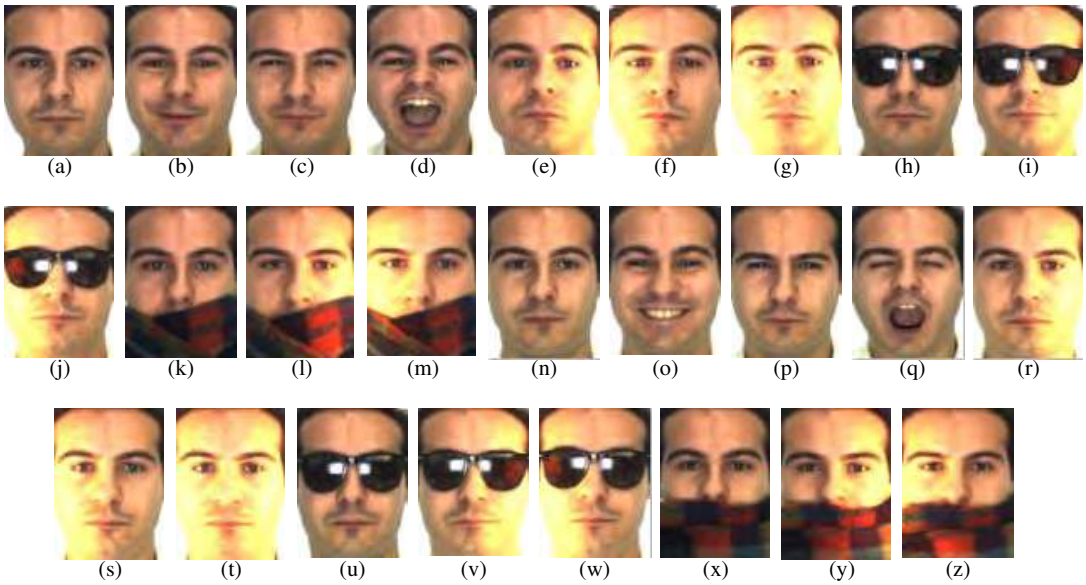

Fig. 4. Sample of images from AR database for 1 subject

FERET database is developed for facial recognition system evaluation. Besides poses, the images in this database have variation of facial expressions and illumination background. It is managed by the Defense Advanced Research Projects Agency (DARPA) and the National Institute of Standards and Technology (NIST). The database is gathered independently by Dr. Harry Wechsler at George Mason University in collaboration with Dr. Philips. The images have been collected in 15 sessions between December 1993 and August 1996. There are a total of 1199 subjects with a total of 14,126 images. However, the number of subjects acquired for system evaluation is 467 subjects. The number of samples for each subject is also not consistent in which the samples ranges from 5 to 90. The files are in PPM format, each of the images is 256×384 pixels in size. Pose variations contained in FERET database is listed in TABLE IV. Examples of the image samples prepared for the following experiments are shown in Fig. 5. In this work, we used 35 subjects only to evaluate the proposed architecture.

Fig. 5. Sample of images from FERET database

B. Preprocessing

Pre-processing stage for AR Purdue database requires RGB to grayscale image converter and image resizing. There are 100 subjects in the database with 26 samples each. The division of training and test samples follows 80/20 ratio that is 2000 and 600 samples respectively. Fig. 6 depicts the pre-processing stage for AR Purdue database.

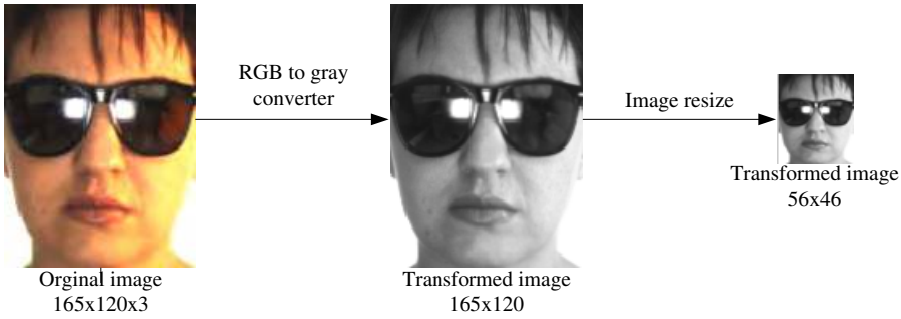

Fig. 6. Pre-processing stages for AR database

FERET database requires additional pre-processing step since the original image is large and includes the subject's body. It requires region-of-interest (ROI) extraction, RGB to grayscale converter and image resizing. The ROI extraction applies Photoscape software to manually crop the images and the remaining pre-processing stages apply MATLAB software. The number of subjects used for design evaluation is 35 subjects with total samples ranges from 12-24 for each subject. Therefore, there are a total of 632 samples with 504 training samples and 128 test samples. This database has Fig. 7 depicts the pre-processing stages for FERET database.

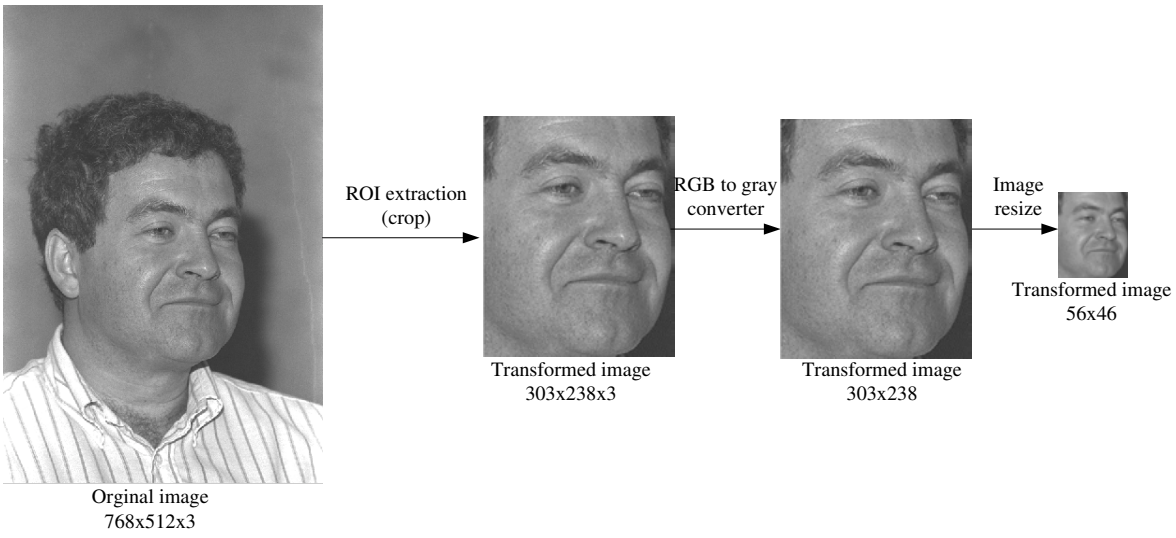

Fig. 7. Pre-processing stages for FERET database

C. Finding the best model: cross-validation results

Images of AR Purdue database consists of variation in light illumination, occlusion (covering part of the face) and variation of facial expressions. The first experiment is conducted to find the best model among seven different architectures using 10-fold cross-validation technique. The architectures to be tested are: *15-45-140*, *15-45-130*, *15-45-120*, *14-45-130*, *16-45-130*, *15-46-130* and *15-44-140* respectively. These numbers represent the number of feature maps at layer C1, C2 and C3. Fig. 8 shows the result of cross-validation process. In this experiment, min-max normalization and Gaussian weight initialization are applied with a zero mean and a standard deviation of 0.03.

As for FERET database, the images contain wide range of poses starting from frontal to 90° poses, variety of facial expressions and illumination effect. In addition, the images are taken in different time periods in which the subjects wear different apparel, different hair style, with or without beard and with or without glasses. The samples contained in this database are more complex than the AR database. Therefore, the architectures to be tested are bigger than the one tested for AR: *14-45-130*, *15-45-130*, *16-45-130*, *15-46-130*, *15-47-130*, *15-48-130*, *15-47-120* and *15-47-140*. Fig. 9 depicts the result of the cross-validation process for the FERET database.

In this experiment, min-max normalization and Gaussian weight initialization are applied with a zero mean and a standard deviation of 0.04.

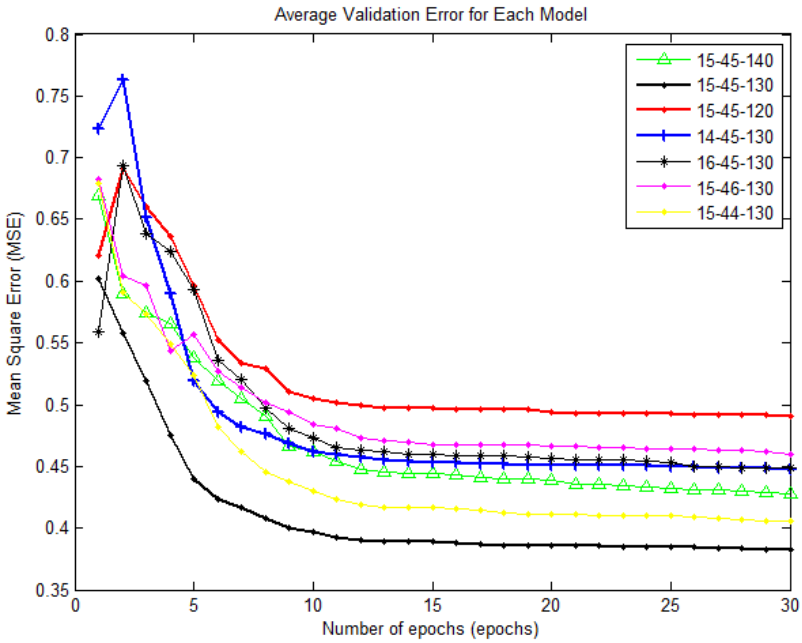

Fig. 8. Cross-validation result for AR database

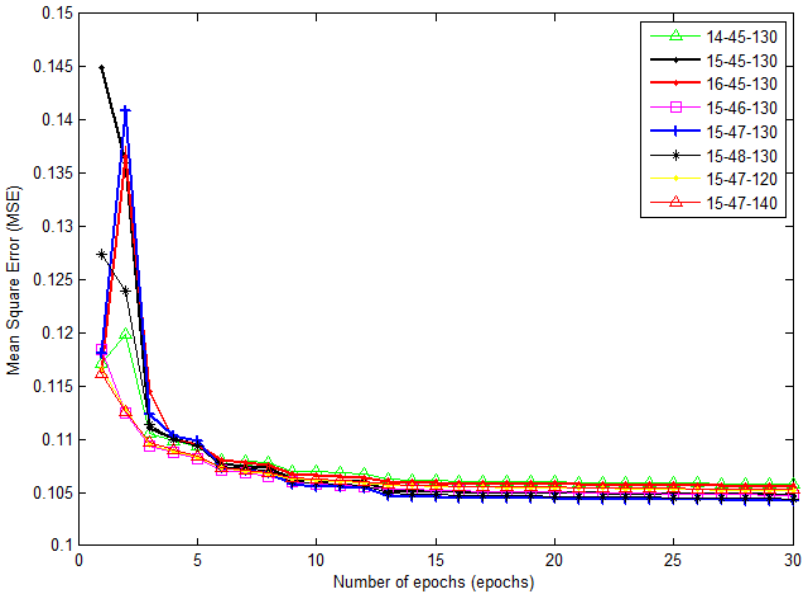

Fig. 9. Cross-validation result for FERET database

As seen in Fig. 8 and Fig. 9, the validation error is stagnant starting from epoch number 20 to 30. Hence, we evaluate the validation error at the 20 epoch point. The lowest validation error is stated by the model *15-45-130* for both of the databases. The performance measure of this generalization ability is based on the percentage of correct classification of the test samples. TABLE IV and TABLE V summarize the performance for AR and FERET database respectively. For AR database, the best accuracy of 96.50% is achieved with the *15-45-130* model. The best accuracy of 81.25% is achieved with the *15-47-130* model for FERET database. This indicates that, the training performance does not necessarily ensure significant performance for test data.

TABLE IV
Performance of the model tested for AR database

Model	Accuracy (%)
15-45-140	95.67
15-45-130	96.50
15-45-120	94.33
14-45-130	95.50
16-45-130	94.50
15-46-130	95.67
15-44-130	96.00

TABLE V
Performance of the model tested for FERET database

Model	Accuracy (%)
14-45-130	67.18
15-45-130	69.53
16-45-130	68.75
15-46-130	67.19
15-47-130	81.25
15-48-130	67.19
15-47-120	68.75
15-47-140	66.41

Once the best model is known, the next step is to identify the best partial connection map between C1 and C2 layer that is also conducted through 10-fold cross-validation process as explained in Section II. The following experiments apply connection table depicted in TABLE II and TABLE III in Section III.

D. Experiment on weight initialization and normalization methods

In this experiment, combinations of different weight initialization and normalization method is evaluated to find the best combination. Four types of weight initialization methods tested are Gaussian, uniform, fan-in and Nguyen-Widrow. For normalization, min-max and Z-score methods are applied. The sample images are normalized to downscale the range of the grayscale image from [0 to 255] to the target range of [-1, +1]. Let x_i and x_i' represent the current pixel value and new value respectively. Then, for min-max normalization, x_i' can be computed by the equation:

$$x_i' = (\max_t - \min_t) * \left[\frac{x_i - \min_v}{\max_v - \min_v} \right] + \min_t \tag{6}$$

where max_t and min_t refers to minimum and maximum range of target, and min_v and max_v indicate the minimum and maximum pixel value of an image. For Z-score normalization method, x_i' is given by the equation:

$$x_i' = \left[\frac{(x_i - \mu_i)}{\sigma_i} \right] \tag{7}$$

where μ_i and σ_i are the mean and standard deviation of the current image. This method produces data with zero mean and a unit variance.

In terms of algorithm, uniform and fan-in weight initialization is almost similar, but they differ in the range of randomization and weights distribution. The range of uniform weights is set to [0.05 to -0.05]. Fan-in initialization uses the range [5.0/fan-in to -5.0/fan-in]. Fan-in is the number of incoming weights into a particular neuron. Gaussian and Nguyen-Widrow methods have similar kind of distribution. Gaussian distribution is determined by the mean and standard deviation, whereas Nguyen-Widrow has a uniform distribution that is scaled by the number of hidden and input neurons, and a constant. The parameter setup for each weight initialization method is set in TABLE VI and TABLE VII for AR and FERET respectively.

TABLE VI
Parameter setup for AR database

Normalization	Weight initialization	Regularization parameter	Constant	Standard deviation
Min-max	Uniform	0.07	0.05	-
	Gaussian	0.07	-	0.03
	Fan-in	0.04	5.0	-
	Nguyen-Widrow	0.07	-	-
Z-score	Uniform	0.05	0.05	-
	Gaussian	0.04	-	0.04
	Fan-in	0.04	5.1	-
	Nguyen-Widrow	0.07	-	-

TABLE VII
Parameter setup for FERET database

Normalization	Weight initialization	Regularization parameter	Constant	Standard deviation
Min-max	Uniform	0.07	0.05	-
	Gaussian	0.07	-	0.04
	Fan-in	0.04	5.0	-
	Nguyen-Widrow	0.07	-	-
Z-score	Uniform	0.05	0.05	-
	Gaussian	0.06	-	0.06
	Fan-in	0.05	1.0	-
	Nguyen-Widrow	0.05	-	-

Training is performed with our selected model (i.e. 15-45-130 model for AR database and 15-47-130 model for FERET database), applying the different combinations of normalization and weight initialization. The best of ten random weight sets for each weight initialization method was chosen for the initial parameters of the network. TABLE VIII presents the results of the experiment, which show that the highest accuracy is achieved in the case when Z-score normalization and Gaussian weight initialization methods are applied for both databases. An accuracy of 99.50% and 85.16% are achieved for AR and FERET database respectively.

TABLE VIII
Accuracy for different combinations of weight initialization algorithm and normalization methods

Normalization	Weight initialization	AR Accuracy (%)	FERET Accuracy (%)
Min-max	Uniform	96.17	73.44
	Gaussian	96.50	82.81
	Fan-in	81.00	70.31
	Nguyen-Widrow	94.83	73.44
Z-score	Uniform	96.17	74.22
	Gaussian	**99.50**	**85.16**
	Fan-in	95.67	71.09
	Nguyen-Widrow	76.12	75.00

Fig. 10 and Fig. 11 illustrate the results for each database after the training process in which the feature maps at each layer behave as a feature detector. From the figure, we can see that some of the feature detectors behave as sharpening, blurring and edge detection filters.

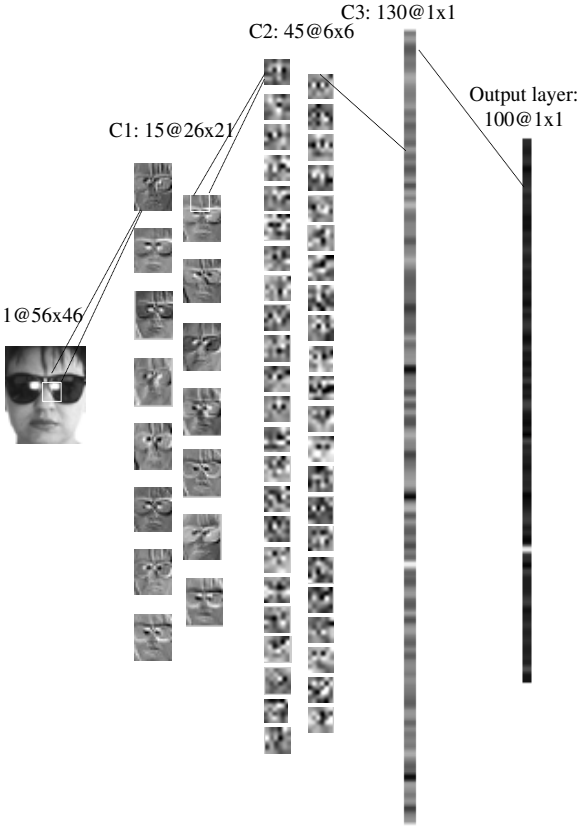

Fig. 10. Feature maps at each layer after the training process for AR database

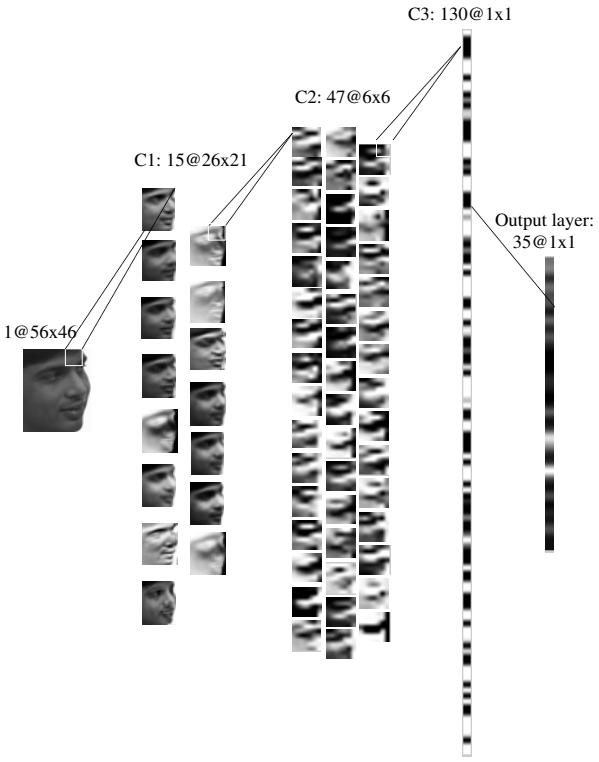

Fig. 11. Feature maps at each layer after the training process for FERET database

E. Benchmarking results

A comparison (TABLE IX) is made for the AR database and the results were found to outperform other approaches [30-32] on the same number of subjects. Rose and Song et al. used more than 100 subjects but have achieved an accuracy of lower than 90%. This shows the superior performance of the proposed approach in handling variation in illumination, occlusions and facial expressions.

By comparing the performance of our proposed approach to other previous work on FERET database (TABLE X), we achieve the third highest accuracy after Rinky et al. and Yaji et al. However, their approach requires specific knowledge of the underlying database. For instance, they have implemented different series of algorithms for different databases. Whenever a new database is applied, a complete re-designing process of the series of algorithms is required. In our CNN approach, we have proven that the same approach could be applied for different databases. Only the number of feature maps at each layer requires adjustment whenever a new database is applied. Shih et al. achieved the highest accuracy of 90.80% with smaller number of subjects. Generally, the accuracy may drop if more number of subjects is applied.

TABLE IX
Benchmarking with other methods for AR database

Reference/Year	Approach	No. of subjects	Accuracy (%)
Roli and Marcialis, 2006 [30]	Semi-supervised PCA	100	85.53
Rose, 2006 [33]	Gabor and log gabor filters	126	89.00
Song et al., 2007 [34]	Parameterized direct LDA	120	<90.00
Jiang et al., 2011 [31]	K-SVD	100	97.80
Patel et al., 2012 [32]	Dictionary-based recognition	100	93.70
Proposed approach	**CNN**	**100**	**99.50**

TABLE X
Benchmarking with other methods for FERET database

Reference/Year	Approach	Notation	No. of subjects	Accuracy (%)
Shih et al., 2005 [35]	Fisherface	fa-rd	20	90.80
Rinky et al., 2012 [36]	DWT	fa-rd	35	88.00
Yaji et al., 2012 [37]	DWT	fa-rd	35	86.45
Proposed approach	**CNN**	fa-re	**35**	**85.71**

V. CONCLUSION

Two new CNN architectures have been proposed. The first architecture handles frontal facial images under occlusions, various illumination and facial expressions. The second CNN architecture tackles facial images with various poses, facial expressions and illumination. Z-score normalization and Gaussian weight initialization algorithm has been found to be the best combination that resulted in the highest accuracy. A classification accuracy of 99.50% and 85.13% are achieved on test samples for AR and FERET database respectively. On a 2.5 GHz Intel i5-3210M quad core processor and 8GB RAM memory, the recognition time taking less than 0.01 seconds is achieved. For future work, the proposed architecture will be extended to improve the accuracy on FERET database and add in new algorithms to include new subjects.

ACKNOWLEDGMENT

This work is supported by the Ministry of Science, Technology and Innovation of Malaysia (MOSTI) and University Technology Malaysia (UTM) under the Technofund grant No.3H001.

REFERENCES

[1] T. Dunstone and N. Yager, Biometric System and Data Analysis: Springer, 2009.
[2] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-Based Learning Applied to Document Recognition," in Proceedings of the IEEE, 1998, pp. 2278-2324.
[3] C. Garcia and M. Delakis, "Convolutional face finder: a neural architecture for fast and robust face detection," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, pp. 1408-1423, 2004.
[4] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a Similarity Metric Discriminatively, with Application to Face Verification," in In Proceedings of CVPR (1) 2005, 2005, pp. 539-546.
[5] T. Fok Hing Chi and A. Bouzerdoum, "A Gender Recognition System using Shunting Inhibitory Convolutional Neural Networks," in International Joint Conference on Neural Networks, 2006, pp. 5336-5341.
[6] F. J. Huang and Y. LeCun, "Large-scale Learning with SVM and Convolutional Nets for Generic Object Categorization," in Proceedings of Computer Vision and Pattern Recognition Conference, 2006.

[7] Ahranjany, Razzazi, Ghassemian, and M.-H. Hassan, "A Very High Accuracy Handwritten Character Recognition System for Farsi/Arabic Digits Using Convolutional Neural Networks," in 2010 IEEE Fifth International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA), 2010.
[8] T. Fok Hing Chi and A. Bouzerdoum, "Texture Classification using Convolutional Neural Networks," in 2006 IEEE Region 10 Conference TENCON, 2006, pp. 1-4.
[9] N. Sun, H.-x. Wang, Z.-h. Ji, C.-r. Zou, and L. Zhao, "An Efficient Algorithm for Kernel Two-Dimensional Principal Component Analysis," Neural Computing and Applications, vol. 17, pp. 59-64, 2008.
[10] X. Wang and X. Tang, "Dual-Space Linear Discriminant Analysis for Face Recognition," in Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004, pp. II-564 - II-569.
[11] D. Omaia, J. v. d. Poel, and L. V.Batista, "2D-DCT Distance Based Face Recognition Using a Reduced Number of Coefficients," in XXII Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI), 2009, pp. 291-298.
[12] D. N. Chandrappa and M. Ravishankar, "Automatic face recognition in a crowded scene using multi layered clutter filtering and independent component analysis," in Intelligent Systems Design and Applications (ISDA), 2012 12th International Conference on, 2012, pp. 552-556.
[13] G. Guo, S. Z. Li, and K. Chan, "Face Recognition by Support Vector Machines," in Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition 2000, pp. 196-201.
[14] L. Zhu and S. Zhu, "Face Recognition based on Orthogonal Discriminant Locality Preserving Projections," Advances in Computational Intelligence and Learning — 14th European Symposium on Artificial Neural Networks 2006, vol. 70, pp. 1543-1546, 2007.
[15] G. Lin and H. De-Shuang, "Human face recognition based on radial basis probabilistic neural network," in Neural Networks, 2003. Proceedings of the International Joint Conference on, 2003, pp. 2208-2211 vol.3.
[16] T. Moe Ma Ma and M. M. Sein, "Multi triangle based automatic face recognition system by using 3D geometric face feature," in Instrumentation and Measurement Technology Conference, 2009. I2MTC '09. IEEE, 2009, pp. 895-899.
[17] A. Z. Pervaiz, "Real Time Face Recognition System Based on EBGM Framework," in Computer Modelling and Simulation (UKSim), 2010 12th International Conference on, 2010, pp. 262-266.
[18] H. Othman and T. Aboulnasr, "Hybrid hidden Markov model for face recognition," in Image Analysis and Interpretation, 2000. Proceedings. 4th IEEE Southwest Symposium, 2000, pp. 36-40.
[19] J. Sung and K. Daijin, "Pose-Robust Facial Expression Recognition Using View-Based 2D + 3D AAM," Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on, vol. 38, pp. 852-866, 2008.
[20] J. F. Pereira, G. D. C. Cavalcanti, and R. Tsang Ing, "Modular Image Principal Component Analysis for face recognition," in Neural Networks, 2009. IJCNN 2009. International Joint Conference on, 2009, pp. 2481-2486.
[21] C. Cunjian, "Decision level fusion of hybrid local features for face recognition," in Neural Networks and Signal Processing, 2008 International Conference on, 2008, pp. 199-204.
[22] L. Yunqi, L. Qingmin, S. Xiaobing, S. Zhenxiang, and C. Dongjie, "3D Face Hierarchical Recognition Based on Geometric and Curvature Features," in Computer Network and Multimedia Technology, 2009. CNMT 2009. International Symposium on, 2009, pp. 1-4.
[23] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back, "Face Recognition: A Convolutional Neural Network Approach," IEEE Transactions on Neural Networks, vol. 8, pp. 98 - 113 1997.
[24] S. Duffner and C. Garcia, "Face recognition using non-linear image reconstruction," in IEEE Conference on Advanced Video and Signal Based Surveillance, 2007, pp. 459-464.
[25] H. Khalajzadeh, M. Mansouri, and M. Teshnehlab, "Hierarchical Structure Based Convolutional Neural Network for Face Recognition," International Journal of Computational Intelligence and Applications, vol. 12, 2013.
[26] P. Y.Simard, D. Steinkraus, and J. C.Platt, "Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis," in Seventh International Conference on Document Analysis and Recognition, 2003, pp. 958-963.
[27] F. Mamalet and C. Garcia, "Simplifying ConvNets for Fast Learning," in 22nd International Conference on Artificial Neural Networks, 2012, pp. 58-65.
[28] Y. LeCun, L. Bottou, B. Orr, and K. Muller, "Efficient BackProp," in Neural Networks: Tricks of the trade, Springer, ed, 1998.
[29] T. Kohonen, "The self-organizing map," Proceedings of the IEEE, vol. 78, pp. 1464-1480, 1990.
[30] F. Roli and G. L. Marcialis, "Semi-supervised PCA-based face recognition using self-training," in Structural, Syntactic, and Statistical Pattern Recognition, ed: Springer, 2006, pp. 560-568.
[31] Z. Jiang, Z. Lin, and L. S. Davis, "Learning a discriminative dictionary for sparse coding via label consistent k-svd," in Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, 2011, pp. 1697-1704.
[32] V. M. Patel, T. Wu, S. Biswas, P. J. Phillips, and R. Chellappa, "Dictionary-based face recognition under variable lighting and pose," Information Forensics and Security, IEEE Transactions on, vol. 7, pp. 954-965, 2012.
[33] N. Rose, "Facial expression classification using gabor and log-gabor filters," in Automatic Face and Gesture Recognition, 2006. FGR 2006. 7th International Conference on, 2006, pp. 346-350.
[34] F. Song, D. Zhang, J. Wang, H. Liu, and Q. Tao, "A parameterized direct LDA and its application to face recognition," Neurocomputing, vol. 71, pp. 191-196, 2007.
[35] F. Shih, C. Fu, and K. Zhang, "Multi-view face identification and pose estimation using B-spline interpolation," Information Sciences, vol. 169, pp. 189-204, 2005.
[36] B. Rinky, P. Mondal, K. Manikantan, and S. Ramachandran, "DWT based Feature Extraction using Edge Tracked Scale Normalization for Enhanced Face Recognition," Procedia Technology, vol. 6, pp. 344-353, 2012.
[37] G. S. Yaji, S. Sarkar, K. Manikantan, and S. Ramachandran, "DWT Feature Extraction Based Face Recognition using Intensity Mapped Unsharp Masking and Laplacian of Gaussian Filtering with Scalar Multiplier," Procedia Technology, vol. 6, pp. 475-484, 2012.