

SpPCANet: a simple deep learning-based feature extraction approach for 3D face recognition

Koushik Dutta¹  · Debotosh Bhattacharjee¹ · Mita Nasipuri¹

Received: 10 August 2019 / Revised: 30 June 2020 / Accepted: 6 August 2020 /

Published online: 20 August 2020

© Springer Science+Business Media, LLC, part of Springer Nature 2020

Abstract

A Sparse Principal Component Analysis Network (SpPCANet) based feature extraction is proposed here for 3D face recognition. The network consists of three basic components: (1) Multistage sparse principal component analysis filters, (2) Binary hashing, and (3) Block-wise histogram computation. Here, the sparse principal component analysis is used to learn multistage filter banks at the convolution stage, which is followed by binary hashing for indexing and block-wise histogram for pooling. Finally, a linear support vector machine (SVM) is used for classifying the features extracted by SpPCANet. The proposed network SpPCANet is a lightweight deep learning network. Three well-known 3D face databases, namely, Frav3D, Bosphorus3D, and Casia3D, are used for validating the proposed system. This proposed network has been extensively studied by varying different parameters, such as the number of filters at the convolution layer and the size of filters at the convolution layer and size of non-overlapping blocks at the pooling layer. Handling all types of variation of faces available in Frav3D, Bosphorus3D, and Casia3D databases, the system has acquired 96.93%, 98.54%, and 88.80% recognition rates, respectively.

Keywords 3D face image · Sparse principal component analysis filter · Binary hashing · Block-wise histogram · Lightweight deep network

✉ Koushik Dutta
koushik.it.22@gmail.com

Debotosh Bhattacharjee
debotoshb@hotmail.com

Mita Nasipuri
mitanasipuri@gmail.com

¹ Computer Science and Engineering, Jadavpur University, 188, Raja S. C. Maulik Road, Kolkata 700032, India

1 Introduction

Convolutional neural network (CNN) or Convnet [34, 42] is currently the most popular tool in computer vision. Various CNN architecture have been designed for image segmentation, feature extraction, classification, and object recognition. Most of the cases, CNN achieves state-of-the-art results on various image databases. In any classification/recognition problem, researchers mainly focus on extracting innovative features, which is one of the part in classification. However, finding innovative features and selecting a more informative subset of them for reducing the complexity of the system are the limitations of hand-crafted features. The learning-based feature extraction approach of CNN has overcome these limitations. In CNN, the key challenges are to design a proper network architecture and choosing the right configuration and parameters such as the number of layers, filter size, choice of pooling function, etc. There exist various convolution network architectures like LeNet-5 [23], AlexNet [22], GoogleNet [34], FaceNet [31], etc.; which are used in computer vision. CNN is recently used in various ways [37] in different applications other than computer vision [38, 39].

Despite the popularity of CNN, the feature learning procedure is not well defined in CNN-based approaches. There exist no mathematical definition of CNN architecture. The architectures mainly follow multiple levels of representation of input data, where higher levels can represent more abstract information of the data. Such an abstract description from learning provides robustness to the classification. The architecture has been divided into some significant stages, where each stage consists of three layers: a convolution filter bank layer, a nonlinear processing layer, and a feature pooling layer. In each stage, the filter bank learns through various techniques like Restricted Boltzmann Machine (RBM) [24], and Regularized Autoencoder [1]. CNN has different variations in different aspects. For example, in some cases, prefixed convolution filters are used at the convolution layer, and there is no need for learning to create filter banks. This is a lightweight representation of the network. Also, these types of convolution architecture have a proper mathematical definition. In [6], the researchers proposed the scattering convolution network (ScatNet) by scattering transformation that computes a translation-invariant representation, using the fixed wavelet filters with a suitable mathematical definition of the filter. The Scatnet performs better than Convnet in handwritten digit recognition and texture discrimination.

Further, the researchers have worked on various other prefixed filter bank-based (FB) approaches such as PCANet [8], DCTNet [27], ICANet [44], etc. The PCANet, an unsupervised deep learning network, achieved good performance in various image classification tasks. PCANet is a straightforward and useful network in multiple domains. In [21], the authors used PCANet for the classification of a scene from a high-resolution remotely sensed image. PCANet is used in [25] for vehicle “make” recognition from the front view of the car image. In [45], the authors have used the PCANet for human age estimation. Gait recognition has been done using curvelet transform and PCANet [10]. The authors extracted features from masked gait energy image using curvelet transform. The extracted features provides new feature space that focuses on covariance property.

Further, PCANet, a simple deeplearning framework, is used for classification. An effective deep learning framework is described in [35], where the stacking of multiple output features is considered and learned through each stage of the Convolutional Neural Network. This network has been used for face recognition. Similar to PCANet, the DCTNet [27] was also proposed for performing face recognition using a DCT filter bank in place of PCA. Another filter bank-based approach, ICANet [44], used independent component analysis (ICA) filters to develop a cascaded linear convolution network for face recognition.

In this work, we have used an extended concept of PCA, i.e., sparse PCA [47], to propose a simple convolutional network model like PCANet. Here, the sparse PCA network (SpPCANet) used sparse PCA filters for developing a 3D face recognition system. According to [47], the abbreviation of sparse principal component analysis is denoted by SPCA. So, maintaining the similarity with the article [47], our proposed network should be abbreviated like SPCANet. But, as already defined, the network in [35] based on stacked principal component analysis is termed as SPCANet in that published article. Now, in our article, we have abbreviated our proposed network as SpPCANet, and from now onwards, in this paper, the "Sparse principal component analysis" would be termed as SpPCA. The SpPCANet architecture consists of three parts: (1) the prefixed filter-based convolutional stage. According to different types of systems, the number of convolution stages may be one or more than one. Here, the single-stage of the convolution network denoted by SpPCANet-1. Further, a convolution network with two stages would be termed as SpPCANet-2, where the output feature map of the first convolution stage is used for training the next convolution stage. (2) After the cascaded convolution, the SpPCA-based convolution image of the final convolution stage transforms into binary, followed by decimal representation within a range of filters. (3) The block-wise histogram is computed on the resultant image to generate feature vectors. Those are considered as the last output features of the proposed convolution network. The process of generating filter banks at the convolution stages is the main difference between PCANet and SpPCANet based recognition systems. Later, in subsection 3.5, the difference is illustrated using the figure on the databases we have considered in this work.

Before detailing the filter-bank generation technique, a brief introduction on sparse PCA is given here. Sparse PCA [47] is an extension of classical PCA. PCA is widely used for dimensionality reduction of the original data. It has numerous applications in computer vision such as human face recognition, handwritten digit recognition, etc. One of the significant drawbacks of PCA is that each principal components are the linear combination of all the original variables, that is, most of the loadings are nonzero. For that, it is difficult to interpret the derived principal components. The covariance matrix of PCA split into scale part (eigenvalues) and direction part (eigenvectors). Loadings of the original data describe as the covariance between the original variables and the unit-scaled components.

The SpPCA using lasso [36] or elastic net [46] regularization technique produces modified principal components with sparse loadings. Sparse PCA improves the interpretability of PCA. For a large number of variables, SpPCA is more consistent than PCA. At first, regression approaches are applied to PCA for optimization, and then sparse loadings are obtained by striking lasso or elastic net constraint on the coefficients of regression. In SpPCA, a good notion of sparsity is rotation invariant. In this work, the system takes the input face images in a 3D point cloud form. At first, we have registered the pose variant 3D faces into frontal pose using the Iterative closest point (ICP) technique. Next, the 2.5D [18] or depth or range images are formed from the 3D point clouds using the mesh-grid technique. In most of the databases, the 3D data contain outlier portions other than the face, so, we have cropped the 2.5D images into a rectangular shape concerning the nose tip point. The resultant 2.5D images are used for the feature extraction using the proposed SpPCANet network. The extracted features are classified using a linear SVM based classifier. The main contributions of this proposed work are given below.

- Calculation of Sparse PCA from the 2.5D depth images
- Creation of SpPCA filters for developing a simple convolution-based network.

The rest of the paper is organized as follows. The details of the methodology given in section 2. In section 3, we present the experimental result and analysis. Finally, section 4 concludes the paper.

2 Proposed methodology

The proposed technique of 3D face recognition using SpPCANet-based filter bank approach consists of four stages, as listed below:

- 3D Face Image Acquisition
- Pre-processing
- Feature Extraction using Sparse Principal Component Analysis Network
- Recognition

2.1 3D face image acquisition

3D data or point cloud is the points in the three-dimensional Cartesian coordinate. The 3D scanner captures the image of an object as a 3D point cloud. There exist various 3D scanners such as structured light scanner, Kinect 3D scanner, laser scanner, etc. In this investigation, three well-known 3D face databases: Frav3D, Bosphorus3D, and Casia3D, are considered as the input of the system. The Minolta Vivid 700 3D laser scanner was used for capturing 3D face data of the Frav3D database. The 3D data of the Bosphorus3D face database was acquired using Inspeck Mega Capturer II 3D structured light scanner. The Minolta Vivid 910 3D scanner employs laser beam light to scan 3D face data of the Casia3D database.

2.2 Pre-processing

The pre-processing consists of three stages, first registration, second 3D depth image creation, and finally, rectangular cropping. In our experiment, we have considered the above mentioned challenging 3D face databases with different variations in faces like pose, expression, and occlusion and combinations of these. In the present work, we have taken a rectangular cropped image as input. To remove the outliers like unwanted portions for the face recognition system, we have cropped the input face in a rectangular shape based on the nose tip as the centroid point of the rectangle. In frontal position, the nose tip point is the nearest point to the 3D camera, and a plane parallel to the XY plane placed at the immediate back of the head is the farthest. So, the nose tip position of the depth face image has the highest depth value. It has been observed that the nose-tip-based automatic cropping technique performs poorly when the faces are in different poses around X-axis, Y-axis, and Z-axis. To overcome the problem, initially, we have registered the pose variant faces with respect to frontal face using the Iterative Closest Point (ICP)-based [4] registration technique. The ICP-based registration uses a raw 3D point cloud. This process tries to find out the rigid transformation that minimizes the least square distance between two points.

After the registration, we have transformed the 3D point cloud of all the frontal and registered faces (including expression and occlusion variation) into depth images. The 3D depth face image is also termed as range image or 2.5D image [18]. It represents a matrix of normalized depth value in the range 0 to 255 (both inclusive). Considering each depth value as an intensity value, the matrix can be represented as a gray-scale image.

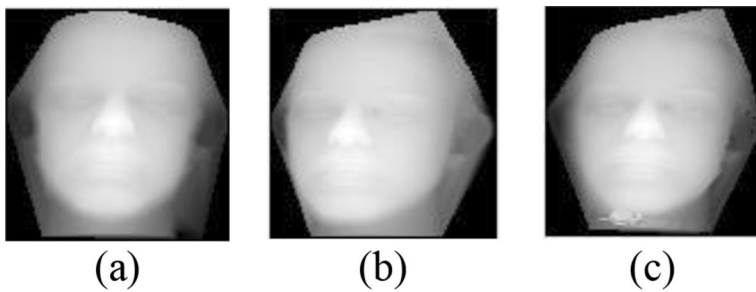


Fig. 1 **a** A frontal 3D depth face image from Frav3D Database; **b** a rotated image; **c** image after registration of 1. (b) w.r.t frontal face

Further, we have cropped all the faces into a rectangular shape according to the said discussion. Figure 1a shows an example of a frontal face image, whereas Fig. 1b represents a pose variant image before registration, and 1c represents its corresponding registered image. Next, Fig. 2 illustrates the process of rectangular cropping, considering the nose tip as the centroid. Figure 2a represents a registered frontal face image, Fig. 2b shows the identification of nose-tip landmark on registered face image and Fig. 2c depicts the rectangular cropped face according to the nose-tip point.

Before cropping, initially, all the depth images of considered three Databases are of equal size 100×100 . In between these three databases, two databases: Frav3D and Casia3D, are considered for cropping operation to discard non-face parts. After cropping, the newly cropped image sizes of Frav3D, and Caisa3D are 81×61 , and 61×41 respectively. We have decided on the height and width of the rectangular cropped image based on the coverage of the significant face portion. The size of the rectangular shape is fully database dependent. The sizes of databases are different due to the outlier portions are not similar. We have only focused on the face portions from the whole image. Later, in the experimental result and analysis section, random images of these three databases are taken to figure out the pre-processing output. The calculation of proper rectangular shape follows most of the subjects of any particular database. The Bosphorus3D database is already well cropped, so, considered the original size of 100×100 as input to the system. The proposed system is not dependent on the size of input data; therefore, we have used different sizes of the input image of different databases.

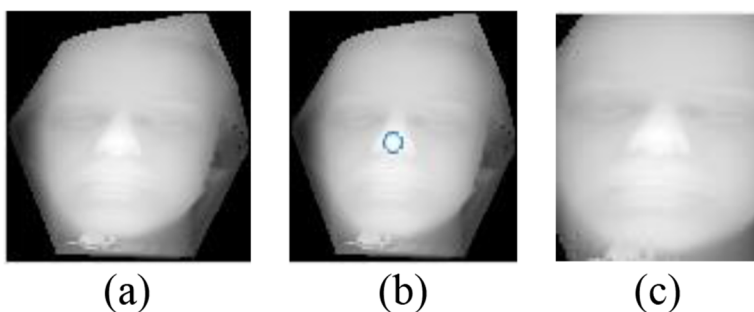


Fig. 2 Process of rectangular cropping of a registered face image

2.3 Feature extraction using sparse principal component analysis network

Like PCANet, the SpPCANet is a lightweight unsupervised convolutional network. The prefixed filter bank-based SpPCANet works on a multistage architecture similar to DCNN but surprisingly gives better performance than DCNN. As already defined, the SpPCANet consists of three parts: convolution layer, nonlinear processing layer, and feature-pooling layer. The architecture of the two-layer filter bank-based sparse PCA network shows in Fig. 3.

From Fig. 3, at the first stage, the original input depth face images convolved by the different convolution filters, i.e., filter bank, the pre-originated filter bank created using the SpPCA technique from the original input images. After the first stage of convolution, in the second stage, again, the same process is repeated, where the convolution filters are created from the convolved image of the first stage. After the completion of the convolution operation, the resultant convolved images are binarized by maintaining a threshold (Heaviside function). After that, a hashing function is used to merge all the output images for individual inputs. Further, a pooling function applied to overlapping or non-overlapping blocks using histogram calculation to create a feature vector for classification. Let us consider the N number of input images I_i , ($i = 1, 2, \dots, N$) of size $m \times n$ and proposed convolution filters are of a size $k_1 \times k_2$. Detailed description of the proposed network for developing SpPCA-based filter from learned input, is divided into two convolution stages, each with six steps, described subsequently.

2.3.1 The first stage of convolution

The first stage of convolution consists of three main steps and six sub-steps under the third step.

Step 1.1: Filter sliding process.

The filter of size $k_1 \times k_2$ is moved over the input face image in an overlapped fashion. The resultant number of block matrices corresponding to the image I_i , is calculated as $s = ((m - k_1) + 1) \times ((n - k_2) + 1)$.

After sliding, reshape each of the block matrices of an image into column vectors and concatenated to a model, denoted as x_{ij} where $j = 1, \dots, s$.

Step 1.2: Centering of data.

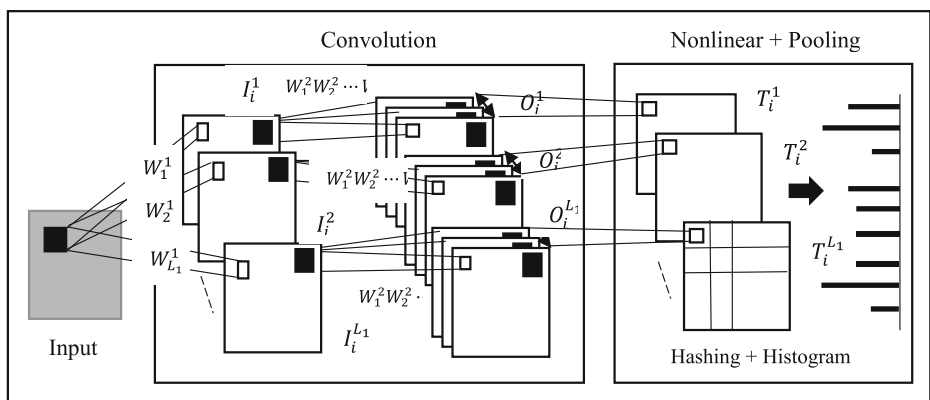


Fig. 3 The architecture of the proposed sparse PCA Network (SpPCANet)

Concatenate each block matrices as a matrix corresponding to the input image I_i , which is denoted as X_i of size $k_1 \times k_2 \times s$ in Eq. (1). Further, subtract the resultant matrices-mean from each matrix corresponding to the input image I_i .

$$X_i = [x_{i,1} \ x_{i,2} \ x_{i,3} \dots x_{i,s}] \quad (1)$$

where $i = 1, 2, \dots, N$ and $\bar{X}_i = X_i - \sum_{i=1}^N X_i$,

Combining all the mean subtracted input training images, the data matrix is represented as Eq. (2).

$$\bar{X} = [\bar{X}_1 \ \bar{X}_2 \dots \bar{X}_N] \quad (2)$$

Step 1.3: Calculation of sparse PCA.

The sparse PCA algorithm is applied to \bar{X}^T , where superscript T represents transpose. Now, the size is $s \times t$, where s and t denote the number of observations and the number of elements, respectively. Here, \bar{X}^T will be denoted as X_{sp} for future use. Without loss of generality, let assume the mean is zero. Otherwise, deduct the mean from each element of the image matrix.

Step 1.3.1: Calculate singular value decomposition (SVD) of X_{sp} as Eq. (3).

$$X_{sp} = UDV^T \quad (3)$$

Where superscript T means transpose, UD are the principal components (PCs) of unit length, and the column vectors of V are the corresponding loadings of the principal components.

Step 1.3.2: Consider first L_1 number of principal components, let $A = V(:, 1 : L_1)$, the loading vectors of first L_1 PCs. To select the sparse PCs, we have calculated the β coefficient to estimate regression. β is an unknown parameter of the regression model. Here, we have used the elastic net regularization technique [46]. Consider a matrix B for storing the values of regularization parameter β . Both A and B are of size $t \times L_1$.

Step 1.3.3: For the fixed $A = [\alpha_1, \dots, \alpha_{L_1}]$, solve β coefficient using an elastic net method [35] for $j = 1, 2, \dots, L_1$ as Eq. (4).

$$\beta_j = \underset{\beta}{\operatorname{argmin}} (\alpha_j - \beta)^T X_{sp}^T X_{sp} (\alpha_j - \beta) + \lambda \|\beta\|^2 + \lambda_{1,j} \|\beta\|_1 \quad (4)$$

For any $\lambda \in [0, 1]$, where $\|\beta\|^2 = (\sum_{i=1}^t \beta_i^2)^{\frac{1}{2}}$ be the l_2 -norm of B , and $\|\beta\|_1 = \sum_{i=1}^t \beta_i$ be the l_1 -norm of B .

Generally, λ is chosen as a small positive integer to overcome potential co-linearity of input data, here X_{sp} .

Step 1.3.4: For a fixed $B = [\beta_1, \dots, \beta_{L_1}]$, compute SVD of $X_{sp}^T X_{sp} B = UDV^T$, after that set $A = UV^T$.

Step 1.3.5: Repeat step 1.3.3 to 1.3.4 until convergence is reached. For convergence, the sum of squares of differences between the sparse loading vectors, estimated in the current and the just previous iterations, is checked against a threshold value. When the difference falls below the threshold, then the convergence is reached. Initially, we have taken a minimum value as a default value of the convergence criterion.

Step 1.3.6: Normalize the sparse, $\hat{V}_j = \frac{\beta_j}{\|\beta_j\|}$, $j = 1, \dots, L_1$.

After calculating the sparse principal component at the first stage, maps the component vectors to matrix formation of size $k_1 \times k_2$ as Eq. (5).

$$W_j^1 = \text{mat}_{k_1 k_2}(\hat{V}_j), j = 1, \dots, L_1 \quad (5)$$

Now, the output of the image after the first convolution as Eq. (6).

$$I_{i,l}^1 = I_i * W_j^1 \quad (6)$$

where $*$ is the convolution operator

2.3.2 The second stage of convolution

The second stage of convolution also includes six steps. The outputs of the first stage of convolution treated as inputs of the second stage. First three steps of the second stage, 2.1, 2.2, and 2.3, same as steps 1.1, 1.2, and 1.3, respectively. Here also, consider the filter of size $k_1 \times k_2$ is slid over the image in the second stage $I_{i,l}^1$, ($i = 1, 2, \dots, N$) and ($l = 1, 2, \dots, L_1$) in an overlapped fashion.

After that, the resultant convolution filters at the second stage denoted by $W_j^2, j = 1, \dots, L_2$. In this stage, L_2 number of filters are used for convolution.

2.3.3 Nonlinear and pooling stage

The nonlinear and pooling stage consists of three steps.

Step 3.1: Binary Encoding

In this nonlinear processing layer, the outputs $\{I_i^l * W_i^2\}_{l=1}^{L_2}$ of the second layer are binarized using a Heaviside step (like) function as Eq. (7).

$$\{H(I_i^l * W_i^2)\}_{l=1}^{L_2} \quad (7)$$

Step 3.2: Decimal conversion

After the binarization process, the resultant output will be one for positive entries and zeros for otherwise. Then, the binarized image is transformed into a single integer-valued image. The transformation function represents L_2 binary bits around each pixel into a single integer as Eq. (8).

$$T_i^l = \sum_{l=1}^{L_2} 2^{l-1} H(I_i^l * W_i^2) \text{ the range of integer is } [0, 2^{L_2}-1], \quad (8)$$

Next, for all the first stage filters, the transformation is denoted as Eq. (9)

$$T = [T_1^1, \dots, T_1^{L_1}, \dots, T_N^1, \dots, T_N^{L_1}] \quad (9)$$

Step 3.3: Histogram Calculation for feature creation

In the pooling stage, the block-based histogram is calculated on T_i^l , where $l = 1, 2, \dots, L_1$. A block size $p_1 \times p_2$ is slid over a resultant decimal image T_i^l with an overlapping ratio equal to 0.

Let, a histogram of resultant M no. of blocks of size $p_1 \times p_2$ transform into a single vector, which is denoted as $Mhist(T_i^l)$ in Eq. (10). After that, the feature has been generated from the input image I_i .

$$feature_i = [Mhist(T_i^1), Mhist(T_i^2), \dots, Mhist(T_i^{L_1})] \quad (10)$$

We have two choices for selecting blocks in a non-overlapping way or overlapping way. It depends on the overlapping ratio. Non-overlapping blocks are mainly used for face recognition, whereas the overlapping blocks are suitable for handwritten digit recognition, texture, and object recognition, etc.

From this discussion, we can conclude that SpPCANet consists mainly of three parameters: Filter size, Number of filters, and Histogram block size. We have used those parameters for our experimental analysis.

2.4 Recognition

After the extraction of features using the learned filter bank of SpPCANet, we have considered linear SVM [13] for their classification. Compared to other kernels of SVM, the linear kernel function takes less time for classification. For recognition, the proposed SpPCA filter banks are applied to input test images for feature extraction, and extracted features are fed to the trained SVM classifier for recognition.

3 Experimental result & analysis

In this section, we explore how the proposed SpPCANet perform in 3D face recognition. Three accessible 3D face databases have been used for this experiment, which are introduced below.

3.1 Experimental databases

Images of various selected subjects from the three considered databases, namely, Frav3D, Bosphorus3D, and Casia3D, have been used to develop the proposed 3D face recognition using SpPCANet. Table 1 Lists these databases and their experimental settings.

The proposed system considers different variations of faces, like pose, expression, and occlusion, as input for the experiment. In this experiment, first, we have registered the pose variant faces using the ICP technique. After that, the images are cropped to remove outliers, if any. Figure 4 shows some of the inputs after registration and rectangular cropping.

Table 1 Description and experimental setting of Frav3D, Bosphorus3D and Casia3D database

Database	Descriptions	No. of class	Cropped Image size	Total no. of images per class
Frav3D [17]	Pose, Expression, Illumination	106	81 × 61	16 (Frontal: 8, Non-frontal: 8)
Bosphorus3D [5]	Pose, Expression, Occlusion	105	100 × 100	49–52 (Frontal: 35–38, Non-frontal: 14)
Casia3D [7]	Poses, Expression, Illumination, combined variations of expressions under illumination and poses under expressions + Smile, laugh, anger, surprise, closed eyes	123	61 × 41	37–38 (Frontal: 17–18, Non-frontal: 20)

Now, we discuss the registration accuracy of the pose variant images. We have calculated root mean square error (RMSE) between the frontal face image and the frontal face obtained after registration of a posed face image of the same subject. The RMSE computes error as the Euclidean distance between the aligned point clouds. Table 2 illustrates the registration accuracy based on different thresholds of RMSE results.

After the completion of the pre-processing stage, the SpPCANet is used to extract features from the pre-processed output. Figure 5 illustrates the output of SpPCANet using pre-processed depth images from the Frav3D Database. In this figure, S-F denotes stage and filter number, where the number of filters in both the convolution stage is two, and the filters are of size 5×5 . Further, in the classification stage, the experiments performed here use 2-fold cross-validation on all of the input databases. All the frontal and registered frontal faces, including expression and occlusion variation images of the input database, have been considered for the experiment. For 2-fold cross-validation, the whole database is equally divided into test and

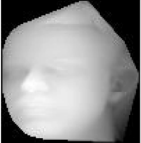
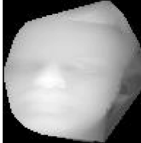





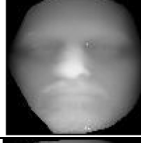
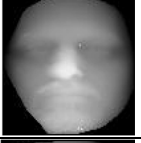


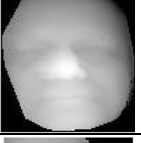
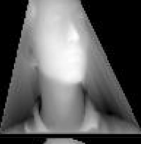
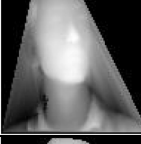
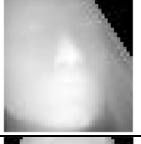



Description of Image	Original Pose-variant Image	Registered Image	Rectangular Cropped Image
Frav3D database (Yaw left rotation)			
Frav3D database (Roll right rotation)			
Bosphorus3D (Roll left rotation)			
Bosphorus3D (Roll left rotation)			
Casia3D database (Yaw right rotation)			
Casia3D database (Pitchup rotation)			

Fig. 4 Example of some pre-processed input images of considered databases

Table 2 Face Registration Accuracy of three individual datasets

Database	RMSE (Threshold ≤ 5)	RMSE (Threshold ≤ 10)
Frav3D	61.25	96.25
Bosphorus3D	31.5	64.25
Casia3D	45.95	71.42

train set randomly. Table 3 given below illustrates the division of training and test set for different experiments of three considered databases.

We analyze the experimental results of recognition performance of the proposed 3D face recognition system with respect to variations of different parameters of the network, like the number of filters, size of filters, histogram block size. An experimental study is presented here to highlight the advantages of SpPCANet over PCANet and Deep Convolution Neural Network (DCNN). A comparison of the proposed technique with some of the previous methods of 3D face recognition is also presented here. Other than this, some ablation experiment is introduced to investigate the effectiveness of different designs of SpPCANet.

3.2 Impact of number of filters on the recognition accuracy

Here, we have analyzed the performance results with the change in the number of filters. Initially, we have considered one-stage of convolution, denoted as SpPCANet-1, where a number of filters L_1 is varied from 5 to 11. When considering two-stage networks, denoted as SpPCANet-2, we have fixed the value of L_2 as 5 and L_1 is varied from 5 to 11. For both cases, we have considered fixed size filters 5×5 ($k_1 \times k_2$), and non-overlapping blocks of size 9×9 ($p_1 \times p_2$). The graphical representation shows the output of the experiments of SpPCANet-1, SpPCANet-2. Figure 6 illustrates the impact of several filters in three considered Databases.

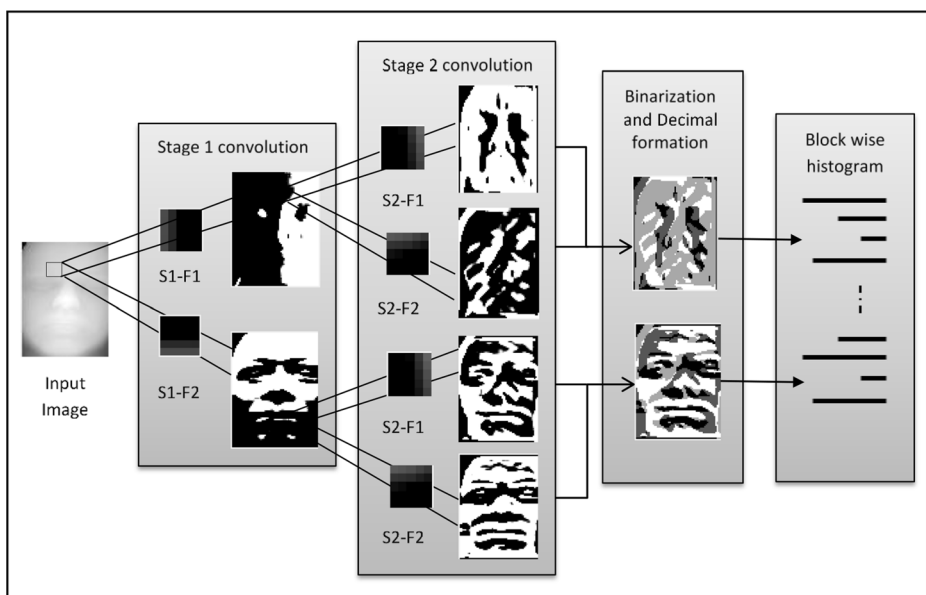


Fig. 5 Output of SpPCANet on 3D depth image of Frav3D database

Table 3 Description of train set and a test set of Frav3D, Bosphorus3D and Casia3D database

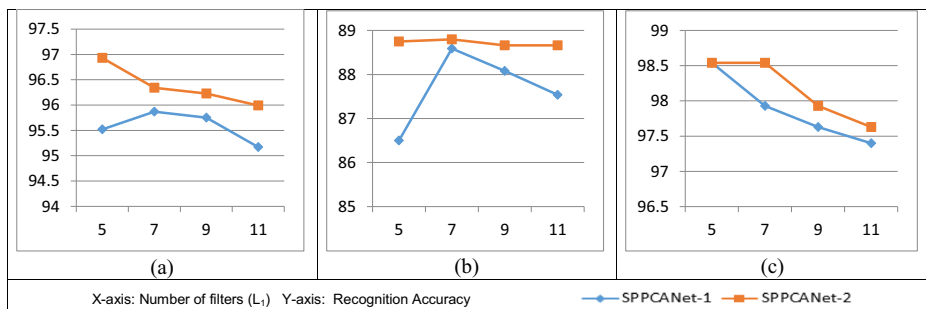
Database	Frontal (with expression and occlusion)			Frontal + Registered frontal (with expression and occlusion)		
	Total no. of image	Train set	Test set	Total no. of image	Train set	Test set
Frav3D	848	424	424	1696	848	848
Bosphorus3D	3289	1668	1621	4725	2403	2322
Casia3D	1968	984	984	4551	2337	2214

3.3 Impact of filter sizes on recognition accuracy

Next, we have examined the impact of different filter sizes on various databases. Figure 7 illustrates the learned SpPCANet 5×5 ($k_1 \times k_2$) filters, where several filters in each layer are $L_1 = L_2 = 5$. Now for the experiment, we have considered two-stage convolution, where the number of filters L_1 and L_2 , in two stages, are fixed as 5, the size of the non-overlapping blocks in two steps are 9×9 ($p_1 \times p_2$), and the size ($k_1 \times k_2$) of three different filters in two stages are 5×5 , 7×7 , and 9×9 , respectively. The impact of different filter sizes illustrates in Fig. 8 graphically on the considered three databases. From the graph, it can be shown that the accuracies are lightly changed with the change of filter sizes for all the databases.

3.4 Impact of histogram block size on recognition accuracy

In the present work, for blocking histogram operation, non-overlapping blocks are used. For analyzing the impact of block size on the recognition performance of the proposed system, we have considered three distinct block-sized: 9×9 , 11×11 , and 13×13 . For this study, we have considered two-stage convolution, where filter size in both stages are 5×5 ($k_1 \times k_2$), and the number of filters is $L_1 = L_2 = 5$. The size ($p_1 \times p_2$) of non-overlapping blocks are changed distinctly: 9×9 , 11×11 , and 13×13 . Figure 9 illustrates the graphical representation of the experiments concerning the change of non-overlapping block size. Similar to Fig. 8, here also, the accuracies are mostly identical to the change of non-overlapping block sizes.

**Fig. 6** Analysis of change of number of filters (L_1) on (a) Frav3D (b) Casia3D (c) Bosphorus3D database

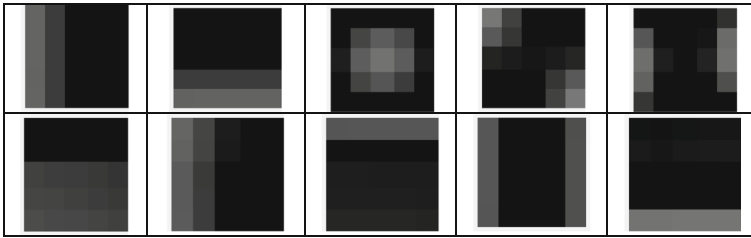


Fig. 7 5×5 SpPCA Filters of two layers

3.5 Impact of number of training samples

We also check the recognition accuracy of the proposed network for different numbers of training samples. However, the numbers of test samples are always the same at the time of testing. Considering 2-fold cross-validation, randomly divided all databases into a nearly equal number of test and train sets. In the case of Frav3D, individually, 848 numbers of images are in test and train set. Now, we have tested the accuracy by taking different numbers of training samples at the time of training. We have chosen only four different numbers of training samples (TS) from 212 to 848 on two networks, where the number of test set samples is 848 for all the cases of training. First, the one-stage network SpPCANet-1, which consists of $L_1 = 5$ filters each of size 5×5 ($k_1 \times k_2$), and the size of the non-overlapping block is 9×9 ($p_1 \times p_2$). Next, two-stage network SpPCANet-2, where the number of filters for both the stages $L_1 = L_2 = 5$ with a fixed size of the filter as 5×5 ($k_1 \times k_2$), and the size of the non-overlapping block is 9×9 ($p_1 \times p_2$). Table 4 illustrates the recognition accuracy for changing the numbers of training samples.

Similarly, we have considered four different numbers of training samples of the Casia3D database. For all the cases, a fixed number of test samples as 2214 are considered for the

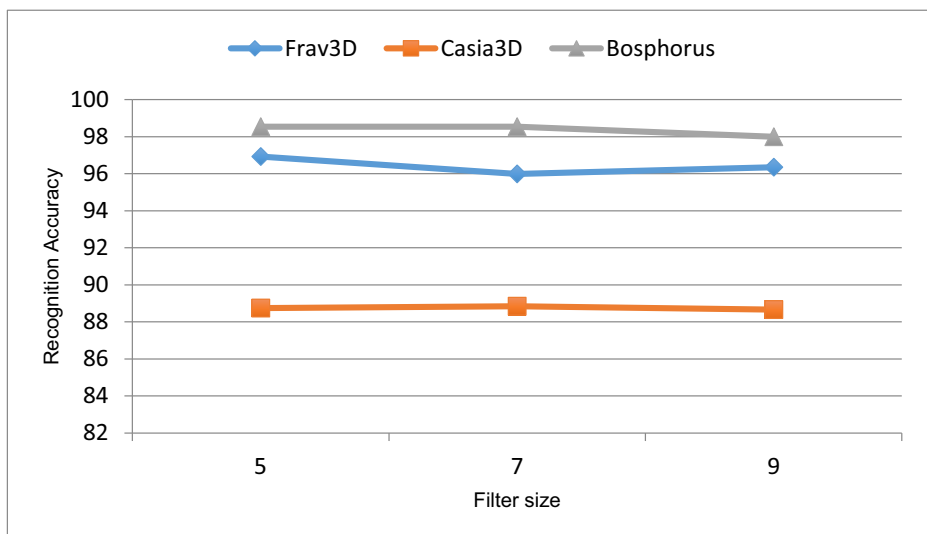


Fig. 8 Analysis of change filter sizes on different databases

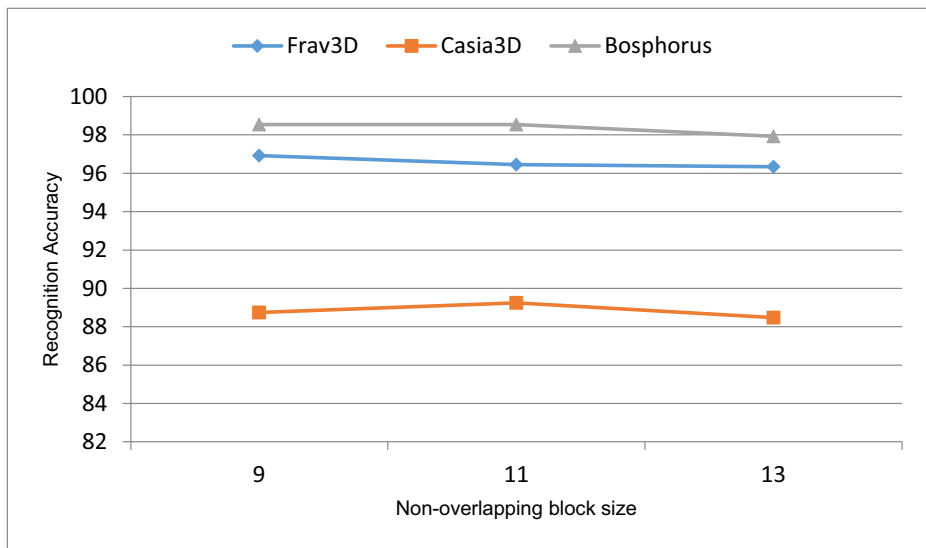


Fig. 9 Analysis of change non-overlapping block sizes on different databases

experiment. Table 5 illustrates the recognition accuracy for changing the numbers of training samples of the Casia3D database.

After the 2-fold cross-validation of the Bosphorus3D database, the numbers of images in the test and train sets are 2322 and 2403, respectively. In this experiment, we have taken four different numbers of training samples, which are from 600 onwards. Table 6 illustrates the recognition accuracy for a different number of training samples.

3.6 Comparison with PCANet and DCNN

We compare the proposed SpPCANet with PCANet and deep convolution neural network (DCNN). From the previous section 3.1, both the networks: PCANet-1 and PCANet-2 are used for comparison with SpPCANet-1 and SpPCANet-2 networks. In [24], the detailed discussion on PCANet is given. As already defined, the difference between PCANet and SpPCANet is the process of creating a filter bank. Figure 10 shows the difference between filter banks, where we have considered all the filters of stage 1, which is produced by training on the Frav3D database using the 5×5 filter size.

Using Fig. 10, we have also represented the stage-1 convolution output images using all convolutional filters of a random face image from the Frav3D database in Fig. 11.

Now, it can be illustrated in Fig. 11 that the first nine filter's outputs will be useful for better classification compare to all the rest of the filter's outputs when the size of the filter is considered 5×5 .

Table 4 Recognition Accuracy (%) for different numbers of training samples of Frav3D database

Network	TS = 212	TS = 424	TS = 636	TS = 848
SpPCANet-1	86.56	89.98	93.28	95.52
SpPCANet-2	87.44	91.27	94.58	96.93

Table 5 Recognition Accuracy (%) for different numbers of training samples of Casia3D database

Network	TS = 615	TS = 1230	TS = 1845	TS = 2337
SpPCANet-1	60.89	69.69	79.45	86.5
SpPCANet-2	61.79	72.63	81.66	88.75

Here, the DCNN-based systems consist of the convolution layer, followed by the relu and max-pooling layer. Finally, the softmax layer is used for classification. The stages of DCNN are like (*Convolution1* → *Relu* → *Maxpooling* → *Convolution2* → *Relu* → *Maxpooling* → *Convolution3* → *Relu* → *Maxpooling* → *Fullyconnected*). The first stage of convolution consists of 256 numbers of filters of size 7×7 , the second stage of convolution consists of 512 numbers of screens of size 5×5 , and the final convolution stage consists of 1024 numbers of filters of size 3×3 . We have not considered any data augmentation techniques for the experiment.

For the comparison process, we have divided all three considered databases into two views: only frontal images with variations and all types of images: frontal and non-frontal registered face images with different variations. Other than pose, the databases consist of neutral, expressive, and occluded faces. The comparison of the results is shown in Table 7.

The filter-bank based approach, SpPCANet, provides better result compared to PCANet and DCNN. According to a study on CNN, CNN gives a better result when the number of training data is enormous. In contrast, the proposed simple convolution network works well and provides a good result with a low amount of training data. Here, the amount of data of all these three 3D face databases are not very large. Within these three databases, the outcomes from the Bosphorus3D database give full 100% accuracy on frontal faces due to high-quality data that acquire by structure light scanner. From the reviews on the Bosphorus3D database, some of the other works [2] [30] used 3D face data of the Bosphorus3D database in their experiment. They produced 100% accuracy on the frontal face with a neutral expression.

The processing times of PCANet, SpPCANet, and DCNN are also different. All the computations have been done on a machine having an Intel Core-i7 processor, 3.6 GHz speed, and 32GB RAM. We have used Matlab version 2018b for the experiment. For time comparison, we have considered the Frav3D database only. The total elapsed time for PCANet-1 (filter size: 5×5 , number of filters: 5, non-overlapping histogram block size: 9×9) is 131.7594 s. Similarly, the total time is taken by SpPCANet-1 is 131.9514 s, whereas the elapsed time of DCNN is around 10 min.

3.7 Comparison with other methods

In this section, we have compared our proposed method with other types of methods based on considered databases. Comparison is made based on two views of the database: first, the frontal pose; second, the registered non-frontal poses. In both the views, we have considered the expression, occlusion variation also. Tables 8, 9, and 10 illustrate the comparison analysis

Table 6 Recognition Accuracy (%) for different numbers of training samples of Bosphorus3D database

Network	TS = 600	TS = 1200	TS = 1800	TS = 2403
SpPCANet-1	70.59	82.73	91.65	98.54
SpPCANet-2	69.51	83.03	91.95	98.54

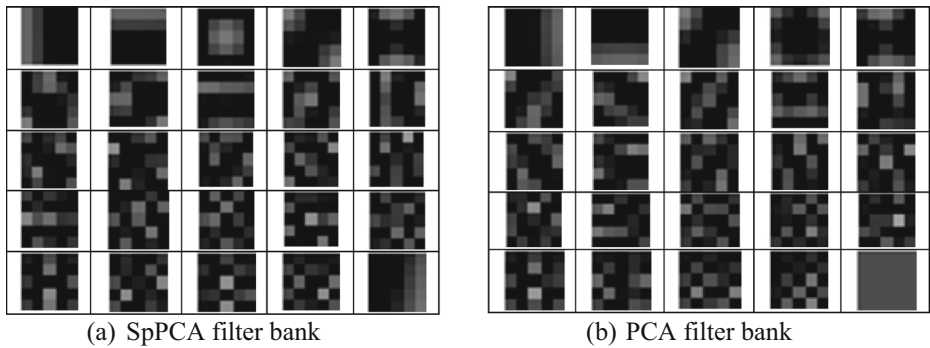


Fig. 10 Differences of filter bank of Frav3D database

with some of the previous works on the three considered databases. Considering the first view, for both the networks: SpPCANet-1 and 2, 100% and nearly 100% recognition performance are acquired in these three databases. Regarding the second view, the recognition accuracies of our proposed method on Frav3D, Bosphorus3D, and Casia3D database are 95.87%, 98.54%, and 88.59%, respectively, using SpPCANet-1 network and 96.93%, 98.54%, and 88.80%,

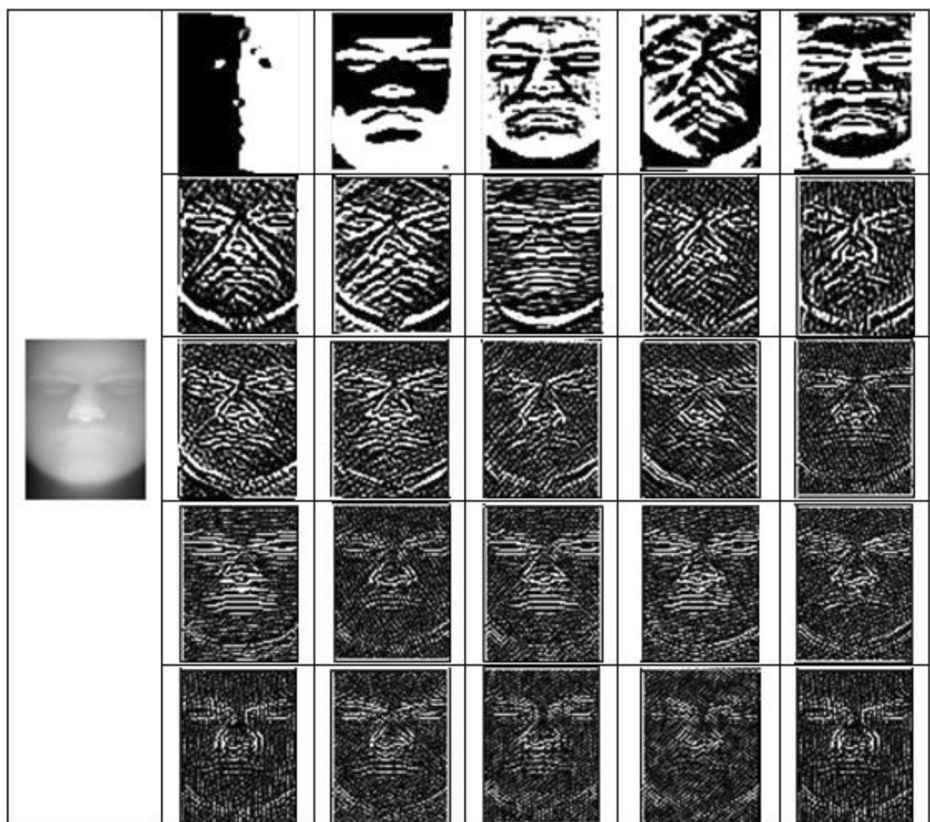


Fig. 11 Convolution output images of input using 25 SpPCA filters of Fig. 10

Table 7 Comparison of Recognition Accuracy (%) between SpPCANet, PCANet, and DCNN on different databases

Technique	Frav3D		Bosphorus3D		Casia3D	
	Frontal	Registered+Frontal	Frontal	Registered+Frontal	Frontal	Registered+Frontal
SpPCANet-1	97.17	95.87	100	98.54	97.97	88.59
SpPCANet-2	98.24	96.93	100	98.54	98.17	88.80
PCANet-1	96.75	95.52	100	97.93	97.97	88.21
PCANet-2	97.99	96.58	100	97.93	97.97	88.66
DCNN	80.19	79.01	81.2	65.09	81.71	65.58

respectively, using SpPCANet-2 network. Compared with other works, it can say, the proposed system is more accurate.

According to these tables, it is shown that the proposed convolution filter-based feature extraction methods, SpPCANet-1 and SpPCANet-2, give better results compared to various state-of-the-art works for all three considered databases. These defined state-of-the-art works are either a local feature or a holistic-based approach. From the last few decades, in most of the cases, the convolution filter-based CNN network gives better recognition accuracies compared to handcrafted feature-based system. The filter-based convolution operation produces the abstract representation of the original input images. The abstract description of the original input image holds more robust features compared to standard handcrafted feature-based systems. In the real-time uncontrolled situation, the filter-based approach performs better using all types of challenges like pose, expression, and occlusion variation of faces.

Moreover, in the CNN based system, there exist some limitations like the requirement of a large number of inputs, high configuration machines to run the convolution network. So, sometimes it is very tough to arrange a high configure machine and large scale of input

Table 8 Comparison of recognition performance of the proposed method with some other method on Frav3D database

Methods	Accuracy (%)		References & Year
	Frontal	Frontal + Non-frontal	
Curvature analysis + SVD + ANN (Classification on the whole face)	86.51	76.08	Ganguly et al. [19], 2014
Multimodal (Fusion of 2D and 3D) + Modular PCA	–	86	Parvathy et al. [29], 2014
ICP-based registration + Surface Normal + KPCA	–	92.25	Bagchi et al. [3], 2015
LBP + HOG + KNN (Region-based classification)	88.86	–	Dutta et al. [15], 2016
ROI detection + Landmark detection + Anthropometric measurement for feature vector creation + SVM	–	95.35	Sghaier et al. [32], 2018
Triangular representation + Volume calculation + KNN	94.28	–	Dutta et al. [16], 2019
Triangular representation + Volume calculation + SVM	95.59	–	Dutta et al. [16], 2019
SpPCANet-1	97.17	95.87	This work
SpPCANet-2	98.24	96.93	This work

Table 9 Comparison of recognition performance of the proposed method with some other method on Bosphorus3D database

Methods	Accuracy (%)		References & Year
	Frontal	Frontal + Non-frontal	
meshDOG descriptor + multi-ring geometric histogram descriptor + keypoint matching for face matching	–	93.4	Werghi et al. [41], 2013
Riemannian framework for analyzing face shape + Statistical shape analysis + Face shape matching	–	89.25	Drira et al. [14], 2013
Multiple Keypoint Descriptor (MKD) + Sparse Representation Classifier (SRC)	98.65	95.03	Zhang et al. [43], 2014
Meshshift + Sparse Representation Classifier (SRC)	96.56	92.99	Zhang et al. [43], 2014
Dual Tree Complex Wavelet Transform (DTCWT) + LDA + NN classifier	–	95.03	Wang et al. [40], 2014
ICP-based registration + Surface Normal + KPCA	–	96.25	Bagchi et al. [3], 2015
Three Fiducial landmark detection using Fully convolution deep network (FCDN) + Adaptive sampling using modified level set speed function + Model-based parameter matching	–	96.3	Gilani et al. [20], 2016
Local directional normal pattern + Patch-based Histogram calculation	–	97.3	Soltanpour et al. [33], 2017
Triangular representation + Volume calculation + KNN	95.21	–	Dutta et al. [16], 2019
Triangular representation + Volume calculation + SVM	96.37	–	Dutta et al. [16], 2019
SpPCANet-1	100	98.54	This work
SpPCANet-2	100	98.54	This work

datasets for developing any system. The proposed SpPCA network, a simple convolution filter-based approach, is useful for less amount of input datasets, and also it can be executed in a low configuration machine.

3.8 Ablation study

To make a comprehensive analysis of the proposed method, we present some ablation experiments to investigate the effectiveness of different designs of the SpPCANet network. Here, different types of ablation experiments have shown in different steps of the proposed system. According to Fig. 3, the steps are mainly focused, such as input, convolution, pooling, and then followed by classification. Specifically, we have analyzed the results by little bit changes in the modules of the actual system and compared the result with the original. For these ablation experiments, we have considered all variations of faces, including the frontal pose of the three considered databases.

As per the initially proposed system, the inputs of the databases are taken without applying any augmentation method in the dataset for increasing the number of inputs. In general, the accuracy of any CNN based system can be increased by augmenting the input datasets. According to Table 7, due to less number of inputs of 3D face databases, the CNN based system is unable to produce better accuracy. Comparatively, our proposed method provides

better accuracy without using any augmentation. Here, we have experimented with the result of our system by incorporating augmentation in our input dataset that used to check the actual need of the augmentation method in our proposed system. Table 11, given below, illustrates the accuracies of the three input databases after applying augmentation and compares the result with the original. There exist various augmentation techniques, here, we have considered simple 2D transformation techniques such as translation of +ve and -ve 5 pixels along with X and Y directions, followed by $+20^\circ$ and -20° rotation of frontal face images. From Table 11, it can be emphasized that the difference between accuracies of with and without augmentation based systems. Augmentation-based systems take extra execution time for augmentation operation.

Next, in the convolution stage, instead of using separate filter banks at each convolution stage, we have used the convolution filter bank of the first stage to all the stages. For doing this, it can be reduced the execution time of our proposed system. Using our defined machine configuration in subsection 3.5, we have checked the training time based on the SpPCANet-2 network. The times taken by the ablation experiment (262.23 s, 823.12 s, 363.64 s) are less compare to the times taken by the original experiment (387.48 s, 1429.65 s, 520.85 s) for Frav3D, Bosphorus3D and Casia3D database. Table 11, given below, illustrates the changes in accuracies for all three databases. This modification is not appropriate for the SpPCANet-1 network. Considering SpPCANet-2, the changes in accuracy do not abruptly decrease, and for some database, it is the same.

Further, at the nonlinear and pooling stage, we have done the modifications in both the non-linear and pooling operation. In this stage, we have suddenly considered these ablation experiments, whether these experiments produce better accuracy or not and also compare execution time. At first, instead of using hashing function in the original architecture, we have simply added

Table 10 Comparison of recognition performance of the proposed method with some other method on Casia3D database

Methods	Accuracy (%)		References & Year
	Frontal	Frontal + Non-frontal	
Extended LBP + Statistical local features+ Normalized correlation distance	94.97	—	Ouamane et al. [28], 2013
TPLBP + Block wise Histogram +SVM	96.48	—	Chouchane et al. [12], 2014
FPLBP + Block wise Histogram +SVM	96.68	—	Chouchane et al. [12], 2014
LPQ + Block wise Histogram +SVM	98.18	—	Chouchane et al. [12], 2014
IPC-based facial area Segmentation + PCA with EFM + SVM	96.75	89.63	Chouchane et al. [11], 2015
Formation of 2D mesh from 3D + combination of Gabor curvature and edge maps +SVM	98.37	—	Torkhani et al. [26], 2017
Combination of Gabor and LTP features + Feature optimization using means of Information Gain Ration + Extreme learning machine (ELM)	98.4	—	Chandrakala et al. [9], 2018
SpPCANet-1	97.97	88.59	This work
SpPCANet-2	98.17	88.80	This work

Table 11 Comparison of recognition accuracies (%) of the proposed method with different ablation experiments of three databases

	Technique	Frav3D	Bosphorus3D	Casia3D
SpPCANet-1	Originally proposed system	95.87	98.54	88.59
	With augmentation	96.1	98.9	91.02
	Using summation followed by normalization instead of the hashing operation	73.79	75.18	68.03
	Using max-pooling instead of histogram calculation	83.25	86.71	68.96
	Classification using nearest neighbour classifier	93.98	92.41	84.09
SpPCANet-2	Originally proposed system	96.93	98.54	88.80
	With augmentation	98.8	99.23	91.14
	Using the same filter-bank for the two convolution stages	96.82	98.54	88.61
	Using summation followed by normalization instead of the hashing operation	77.28	77.08	71.98
	Using max-pooling instead of histogram calculation	88.32	89.34	73.39
	Classification using nearest neighbour classifier	95.66	92.84	84.87

the pixels' values of final convolution matrices of an input face. Further, we have normalized the added values in between 0 to 2^{η} , where η = number of filters. After that, as per the original architecture, calculate the histograms of distinct blocks on the resultant matrix. The rest of the operation is similar to the original. In the second approach, we have used max-pooling operation on separate blocks instead of histogram calculation. Table 11 illustrates the accuracies of both experimental operations. According to Table 11, there are havoc changes in the accuracies of all datasets in both approaches. Similar to the previous stage, we have also checked the execution time, the application of this ablation experiment reduces the time compared to the use of hashing function. Considering the SpPCANet-2 network, the execution times of the nonlinear and pooling stages are considered for time and compared with the original system for all the databases. In the case of Frav3D, Bosphorus3D, and Casia3D database, the execution times of the nonlinear and pooling stages of the first ablation experiment are 0.19 s, 0.27 s, 0.12 s and 0.22 s, 0.33 s, 0.15 s for second ablation experiment of this stage. Now, the execution times of the nonlinear and pooling stages of the original system are 0.22 s, 0.33 s, and 0.16 s. Though the time does not change, the accuracies are reduced more for both the approaches.

Finally, we have done modifications in the classification stage. Instead of using SVM for classification in our proposed SpPCANet system, we have considered a simple nearest neighbor (NN) classifier for a comprehensive analysis of accuracy. Here, we have checked the differences in accuracy and time. Table 11 illustrates that SVM gives better accuracy than NN. Further, we have reviewed the time differences between the training time of SVM and NN on the SpPCANet-2 network. Using Frav3D, Bosphorus3D, and Casia3D databases, SpPCANet-2 network, the training times taken by SVM are 5.45 s, 27.92 s, and 7.06 s, whereas the training times taken by NN are 3.92 s, 22.11 s, and 2.06 s. Though the training times taken by NN are less compare to SVM, but accuracies are not improved for all three databases.

4 Conclusions and future work

In this work, the most straightforward unsupervised convolution learning network, SpPCANet, has been used for feature extraction followed by 3D face recognition. Compared to DCNN, the SpPCANet model is simple, and it gives better accuracy on fewer amounts of input data of the

considered database. SpPCANet has three parts: a two-stage convolutional layer, a binary hash, and block histogram processes. It also provides the mathematical analysis and justification. Like DCNN models, all settings of the parameters: number of filters, size of the filters, and histogram block size depending on the accuracy of the system. From the experimental analysis section, it is clear that the accuracies are not too much varied in any scenario for different databases. Mainly, the large size of histogram blocks covers a large area of the face that maintain the robustness of the system with respect to local deformation such as facial expression and occlusion. According to our work, we have considered three well-known challenging databases for maintaining the reliability of the system. Here, we have maintained the same settings of those parameters for working with these three datasets, and the system produces good accuracy for all the databases in comparison with previous state-of-the-art works of these same databases. Fine-tuned of these parameters leads to good accuracy of the system. In the convolution stage, two stages of convolution are sufficient for recognition; from most of the cases, more than two stages of convolution give the same accuracy as in stage 2, and it uses more memory for computing and also consume more time. The SpPCANet also performs better compared to various other handcrafted features based methods on the considered three databases. Initially, the SpPCANet has tested on three accessible 3D face databases. Further, we will apply the network for other 3D face databases. We will also use it for different types of works like handwritten text recognition, activity recognition, etc.

Acknowledgments The first author is grateful to the Ministry of Electronics and Information Technology (MeitY), Govt. of India, for the grant of the Visvesvaraya doctorate fellowship award. The authors are also thankful to CMATER laboratory of the Department of Computer Science and Engineering, Jadavpur University, Kolkata, India, for providing the necessary infrastructure for this work.

References

1. Alain G, Bengio Y (2014) What regularized auto-encoders learn from the data-generating distribution. *J Mach Learn Res* 15:3743–3773
2. Alyuz N, Gokberk B, Dibeklioglu H, Savran A, Salah AA, Akarun L, Sankur B (2008) 3D Face Recognition Benchmarks on the Bosphorus3D Database with Focus on Facial Expressions. In: Schouten B, Juul NC, Drygajlo A, Tistarelli M (eds) *Biometrics and Identity Management. BioID 2008. Lecture notes in computer science*, vol 5372. Springer, Berlin Heidelberg, pp 57–66
3. Bagechi P, Bhattacharjee D, Nasipuri M (2015). 3D face recognition using surface Normals. In: *Proc. IEEE region 10 conference, TENCON - 2015*. <https://doi.org/10.1109/TENCON.2015.7372819>
4. Besl PJ, McKay ND (1992) A method for registration of 3-D shapes. *IEEE trans. Pattern anal. Mach. Intell. (T-PAMI)* 14(2):239–256
5. BOSPHORUS3D3D: <http://bosphorus.ee.boun.edu.tr/default.aspx>
6. Bruna J, Mallat S (2013) Invariant scattering convolution networks. *IEEE trans. Pattern anal. Mach. Intell. (T-PAMI)* 35(8):1872–1886
7. CASIA3D: <http://www.idealtest.org/dbDetailForUser.do?id=8>
8. Chan TH, Jia K, Gao S, Lu J, Zeng Z, Ma Y (2015) PCANet: a simple deep learning baseline for image classification? *IEEE Trans Image Process* 24(12):5017–5032
9. Chandrakala M, Ravi S (2018) Effective 3D face recognition technique based on Gabor and LTP features. *International Journal of Engineering and Advanced Technology (IJEAT)* 8(2S):284–290
10. Chhatrala R, Jadhav D (2017) Gait recognition based on curvelet transform and PCANet. *Pattern Recog. Image Anal* 27(3):525–531. <https://doi.org/10.1134/S1054661817030075>
11. Chouchane A, Belahcene M (2015) 3D and 2D face recognition using integral projection curves based depth and intensity images. *Int J Intell Syst Technol Appl* 14(1):50–69
12. Chouchane A, Belahcene M, Ouamane A, Bourennane S (2014) 3D face recognition based on histograms of local descriptors. In: *Proc. 4th international conference on image processing theory, tools and applications (IPTA)*, Paris, France. <https://doi.org/10.1109/IPTA.2014.7001925>
13. Cortes C, Vapnik V (1995) Support-vector networks. *Mach Learn* 20(3):273–297

14. Drira H, Amor BB, Srivastava A, Daoudi M, Slama R (2013) 3D face recognition under expressions, occlusions and pose variations. *IEEE trans. Pattern anal. Mach. Intell. (T-PAMI)* 35(9):2270–2283
15. Dutta K, Bhattacharjee D, Nasipuri M (2016). Expression and occlusion invariant 3D face recognition based on region classifier. In: *Proc. 1st international conference on information technology, information systems and electrical engineering (ICITISEE)*, pp. 99–104. <https://doi.org/10.1109/ICITISEE.2016.7803055>
16. Dutta K, Bhattacharjee D, Nasipuri M (2019) 3D face recognition based on volumetric representation of range image. In: Chaki R, Cortesi a, Saeed K, Chaki N (eds) *advance computing and Systems for Security. Advance in Intelligent Systems and Computing* 883:175–189. https://doi.org/10.1007/978-981-13-3702-4_11
17. FRAV3D: <http://www.frav.es/databases>
18. Ganguly S, Bhattacharjee D, and Nasipuri M (2014). 2.5D face images: acquisition, processing and application. In *Proc. ICC 2014 -computer networks and security*, pp. 36–44
19. Ganguly S, Bhattacharjee D, Nasipuri M (2014) 3D face recognition from range images based on curvature analysis. *ICTACT Journal on image and video processing* 4(3):748–753. <https://doi.org/10.21917/ijivp.2014.0108>
20. Gilani SZ, Mian A (2016). Towards large-scale 3D face recognition. *International Conference on Digital Image Computing: Techniques and Applications (DICTA)*
21. Huang D, Du Y, He Q, Song W, Liu K (2016) Scene classification in high resolution remotely sensed images based on PCANet. *Web Technologies and Applications, APWeb, Springer, Cham* 9865:179–190. https://doi.org/10.1007/978-3-319-45835-9_16
22. Krizhevsky A, Sutskever I, Hinton GE (2012) ImageNet classification with deep convolutional neural networks. In: *Proc. 25th international conference on neural information processing systems. Lake Tahoe, Nevada*, pp 1097–1105
23. LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. In: *Proc of the IEEE* 86(11):2278–2324
24. Lee H, Grosse R, Rananath R, and Ng A Y (2009). Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In: *Proc. 26th Annu. ICML*, pp. 609–616. <https://doi.org/10.1145/1553374.1553453>
25. Li B, Dong Y, Zhao D, Wen Z, and Yang L (2016). A PCANet based method for vehicle make recognition. In: *Proc. 19th international conference on intelligent transportation systems (ITSC)*, IEEE. Pp. 2404–2409. <https://doi.org/10.1109/ITSC.2016.7795943>
26. Li C, Tan Y, Wang D, Ma P (2017) Research on 3D face recognition method in cloud environment based on semi supervised clustering algorithm. *Multimed Tools Appl* 6:17055–17073
27. Ng CJ, and Teoh ABJ (2015). DCTNet: a simple learning-free approach for face recognition. In: *Proc. APSIPA*, pp. 761–768. <https://doi.org/10.1109/APSIPA.2015.7415375>
28. Ouamane A, Belahcene M, Bourennane S (2013). Multimodal 3D and 2D face authentication approach using extended LBP and statistic local features proposed. In: *Proc. European workshop on visual information processing (EUVIP)*, pp. 130–135
29. Parvathy SB, Naveen S, Moni RS (2014). A novel approach for multimodal face recognition system based on modular PCA. In: *proc. 1st international conference on computational systems and communications (ICCS)*, pp. 127–132. <https://doi.org/10.1109/COMPSC.2014.7032634>
30. Ratyal N, Taj IA, Sajid M, Mahmood A, Razzaq S, Dar SH, Ali N, Usman M, Baig MJA, Mussadiq U (2019). Deeply learned pose invariant image analysis with applications in 3D face recognition. *Mathematical problems in engineering*. <https://doi.org/10.1155/2019/3547416>
31. Schroff F, Kalenichenko D, and Philbin J (2015). FaceNet: a unified embedding for face recognition and clustering. In: *Proc IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* <https://doi.org/10.1109/CVPR.2015.7298682>
32. Sghaier S, Farhat W, Souani C (2018) Novel technique for 3D face recognition using anthropometric methodology. *International Journal of Ambient Computing and Intelligence* 9(1):60–77
33. Soltanpour S, Wu QMJ (2017). High-order local Normal derivative pattern (LNDP) for 3D face recognition. *International conference on image processing (ICIP)*. Pp. 2811–2815
34. Szegegy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, and Rabinovich A (2015). Going deeper with convolutions. In: *Proc IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* <https://doi.org/10.1109/CVPR.2015.7298594>
35. Tian L, Fan C, Ming Y (2015). Stacked PCA network (SPCANet): an effective deep learning for face recognition. In: *Proc. IEEE International Conference on Digital Signal Processing*, pp. 1039–1043. <https://doi.org/10.1109/ICDSP.2015.7252036>
36. Tibshirani R (1996) Regression shrinkage and selection via the lasso. *J R Stat Soc Ser B Methodol* 58(1): 267–288 <https://www.jstor.org/stable/2346178>

37. Tong M, Chen Y, Zhao M, Bu H, Xi S (2019) A deep discriminative and robust nonnegative matrix factorization network method with soft label constraint. *Neural Comput & Applic* 31:7447–7475. <https://doi.org/10.1007/s00521-018-3554-6>
38. Tong M, Li M, Bai H, Lei M, Zhao M (2020) DKD–DAD: a novel framework with discriminative kinematic descriptor and deep attention-pooled descriptor for action recognition. *Neural Comput & Applic* 32:5285–5302. <https://doi.org/10.1007/s00521-019-04030-1>
39. Tong M, Zhao M, Chen Y, Wang H (2019) D3-LND: a two-stream framework with discriminant deep descriptor, linear CMDT and nonlinear KCMDT descriptors for action recognition. *Neurocomputing* 325: 90–100. <https://doi.org/10.1016/j.neucom.2018.09.086>
40. Wang X, Ruan Q, Jin Y, and An G (2014). Three-dimensional face recognition under expression variation. *EURASIP Journal on Image and Video Processing* <https://doi.org/10.1186/1687-5281-2014-51>, 2014
41. Werghi N, Berretti S, Bimbo A D, Pala P (2013). Local descriptors matching for 3D face recognition. In: *Proc. IEEE international conference on image processing (ICIP)*, Australia, pp. 3710–3714. <https://doi.org/10.1109/ICIP.2013.6738765>
42. Zeiler MD, Fergus R (2014) Visualizing and Understanding Convolutional Networks. In: Fleet D, Pajdla T, Schiele B, Tuytelaars T (eds) *Computer Vision – ECCV 2014*. Lecture notes in computer science. Springer, Cham. https://doi.org/10.1007/978-3-319-10590-1_53
43. Zhang L, Ding Z, Li H, Shen Y, Lu J (2014) 3D face recognition based on multiple Keypoint descriptors and sparse representation. *PLoS One* 9(6):e100120. <https://doi.org/10.1371/journal.pone.0100120>
44. Zhang Y, Geng T, Wu X, Zhou J, and Gao D (2018). ICANet: a simple cascade linear convolution network for face recognition. *EURASIP Journal on Image and Video Processing* <https://doi.org/10.1186/s13640-018-0288-4>, 2018
45. Zheng D, Du J, Fan W, Wang J, Zhai C (2016) Deep learning with PCANet for human age estimation. In: Huang DS, Jo KH (eds) *international conference on intelligent computing (ICIC)*, lecture notes in computer science, springer. Cham. 9772:300–310. https://doi.org/10.1007/978-3-319-42294-7_26
46. Zou H, Hastie T (2005) Regularization and variable selection via the elastic net. *J R Stat Soc Ser B (Stat Methodol)* 67(2):301–320 <https://www.jstor.org/stable/3647580>
47. Zou H, Hastie T, Tibshirani R (2006) Sparse principal component analysis. *J Comput Graph Stat* 15(2): 265–286. <https://doi.org/10.1198/106186006X113430>

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Koushik Dutta is a Visvesvaraya PhD-fellow in the Department of Computer Science and Engineering, Jadavpur University, Kolkata, India. He received his M.Tech degree in Information Technology from Indian Institute of Engineering Science and Technology, Shibpur, Howrah, India in 2014. His current research interest is 3D face Recognition using 2.5D image.



Debotosh Bhattacharjee is working as a full professor in the Department of Computer Science and Engineering, Jadavpur University with fourteen years of post-PhD experience. His research interests pertain to the applications of machine learning techniques for Face Recognition, Gait Analysis, Hand Geometry Recognition, and Diagnostic Image Analysis. He has authored or coauthored more than 250 journals, conference publications, including several book chapters in the areas of Biometrics and Medical Image Processing. Two US patents have been granted on his works. Prof. Bhattacharjee has been granted sponsored projects by the Govt. of India with a total amount of around INR 2 Crore.



Mita Nasipuri received her B.E.Tel.E., M.E.Tel.E, and Ph.D. (Engg.) degrees from Jadavpur University, in 1979, 1981 and 1990, respectively. Prof. Nasipuri has been a faculty member of J.U since 1987. Her current research interest includes image processing, pattern recognition, and multimedia systems. She is a senior member of the IEEE, U.S.A., Fellow of I.E. (India) and W.B.A.S.T, Kolkata, India.