



## Implementation of PCA for Recognition of Hand Gesture Representing Alphabets

Ashis Pradhan\*, Shubham Kumar, Dependra Dhakal, Bishal Pradhan

CSE Department, SMIT, Sikkim Manipal University,  
Sikkim, India

**Abstract**— Establishment of communication between two peers plays a crucial role in knowledge exchange. It can be done either using linguistic features or gestures. The development of procedure for realizing gestures into meaningful information plays a pivotal role in instances where linguistic feature cannot be used. In such situations gestures can be used as the alternative for conveying the same. This paper basically aims at discussing implementation of Principal Component Analysis (PCA) for recognizing hand gesture that represents alphabets. PCA is a useful statistical technique that has found application in field such as image recognition, dimension reduction. It identifies pattern by reducing the dimension of data with minimum loss of information.

**Keywords**—Information exchange, Hand Gesture, Pattern Recognition, Dimension reduction, PCA

### I. INTRODUCTION

Standard sign languages (SL) are known as Deaf and Dumb languages. SLs are gestural languages which contain symbolic encoded messages for communication without speech channel. They are unique in some ways in that they cannot be written like spoken languages. Sign languages vary from country to country with its own vocabulary and grammar. Even within one country, sign language can vary from region to region like spoken languages. American Sign Language (ASL) is a language used by American deaf and dumb community.

Gestures are bodily actions made by humans to convey meaningful information to others. It comes easily to human and so using these gestures as a mode of interaction will help humans to interact with computers easily. A gesture is scientifically categorized into two distinctive categories: dynamic and static. A dynamic gesture is intended to change over a period of time whereas a static gesture is observed at the spurt of time. A waving hand means goodbye is an example of dynamic gesture and the stop sign is an example of static gesture. In this project we mainly focus on static hand gestures. The primary goal of hand gesture recognition system is to create a system which can identify specific hand gestures and use them to convey information.

A digital image is composed of pixels, where a pixel is the smallest unit of image. Each pixel represents intensity value at that point in an image and this value depends on the type of image.

There are mainly three types of digital images. The digital image whose intensity value of each pixel is a value ranging between black to white called Black and White Image (Grayscale). While the digital image whose each pixel is contributed by different colors based on color scheme (RGB, HSV) used called Color Image whereas the digital image in which the value of each pixel is represented by one bit (either 0 or 1) only is called Binary Image.

PCA is a statistical procedure concerned with elucidating the covariance structure of a set of variables. In particular it allows one to identify the principal directions in which the data varies. It extracts principal features of an image. These features can principally differentiate among various input images. This technique produces results in fast and relatively more accurate manner. Basically PCA method is based on a concept in which the database images are represented using Eigen value and similarly, the Eigen value is found for test image also. Finally, based on the Euclidean distance of these Eigen values, the images were correctly classified. In linear algebra, the eigenvectors of a linear operator are non-zero vectors which, when operated by the operator, result in a scalar multiple of them. Scalar is then called Eigen value ( $\lambda$ ) associated with the eigenvector ( $X$ ). Eigen vector is a vector that is scaled by linear transformation. It is a property of matrix. When a matrix acts on it, only the vector magnitude is changed not the direction.

$$AX = \lambda X, \text{ where } A \text{ is a vector function.}$$

$$(A - \lambda I)X = 0, \text{ where } I \text{ is the identity matrix.}$$

This is a homogeneous system of equations and form fundamental linear algebra. We know a non-trivial solution exists if and only if

$$\text{Det}(A - \lambda I) = 0, \text{ where Det denotes determinant.}$$

When evaluated becomes a polynomial of degree  $n$ . This is called characteristic polynomial of  $A$ . If  $A$  is  $N$  by  $N$  then there are  $n$  solutions or  $n$  roots of the characteristic polynomial. Thus there are  $n$  Eigen values of  $A$  satisfying the equation.

$$AX_i = \lambda_i X_i, \text{ where } i = 1, 2, 3, \dots, n$$

If the Eigen values are all distinct, there are  $n$  associated linearly independent eigenvectors, whose directions are unique, which span an  $n$  dimensional Euclidean space.

This paper basically aims at studying the existing system that recognizes hand gestures and implements the Principal Component Analysis (PCA) algorithm for recognizing bare hand gesture representing English alphabets. Furthermore, the designed system is be analysed to check for rotation invariant property of an algorithm.

The basic gestures that the system will identify are the following standardized gestures:



Figure 1: American Sign Language (ASL) representing Alphabets

The system will include a database of low-resolution images representing hand gestures and follows an algorithm that processes the input images and then classifies the hand gestures correctly. It will contain a setup procedure that executes once, where the algorithm is trained on given image set that contains different hand gestures. After the setup done, the system will be able to classify the given hand gesture, as input, based on the features extracted defined during the training time. One can primarily work on PCA (Principal Component Analysis) in which the principal components that is eigenvector and eigenvalue of every image is found and stored in the database. These components are then compared with the input's components.

## II. METHODOLOGY USED

Take images for your database from at least three different hands which comprises of gestures of alphabets. Create a text file named as handlist.txt which contains path and name of each image.

For example:

```
Database\A.JPG A
Database\A1.JPG A
Database\A2.JPG A
Database\B.JPG B
Database\B1.JPG B
Database\B2.JPG B
```

Following steps are taken to implement PCA:

1. Read the list of images and store it in an array say 'handlist', so handlist contains:  
A Database\A.JPG  
A1 Database\A1.JPG And so on...
2. Size of handlist (numoffile) =78
3. Read all images and store the pixel value of each images in a 2D array in column wise fashion.

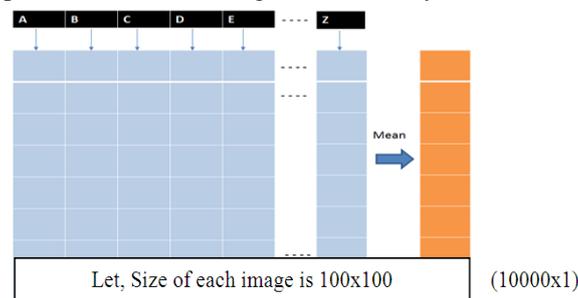
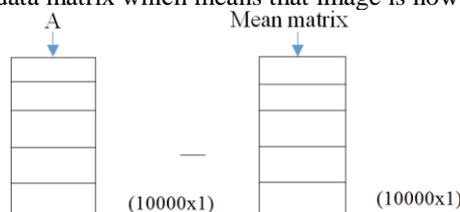


Figure 2: Representation of pixel storage of images

4. Calculate the mean of each row to get an average image and store that result in 'mn'. Size of 'mn' matrix will be (10000x1).
5. Now calculate the mean centered data matrix which means that image is now shifted to zero or an average image.



Similarly calculate mean centered data matrix for all alphabets and store it in 'sd' matrix. Therefore size of 'sd' will be 10000 x 78, where each column represents an alphabet.

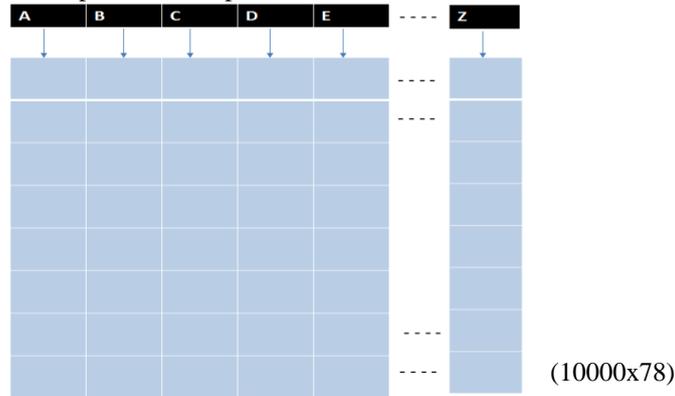


Figure 3: Representation of mean centered data matrix.

6. Calculate covariance matrix using formula (dataset' x dataset) / n and store the result in a new matrix named as 'covariance'. Size of 'covariance':

$$\begin{matrix} & \text{dataset}' & \times & \text{dataset} & = & \text{covariance} \\ \swarrow & & & \downarrow & & \swarrow \\ (78 \times 10000) & & & (10000 \times 78) & & (78 \times 78) \end{matrix}$$

Characteristics of covariance matrix:

- a) Diagonal elements of the covariance matrix are the variance of each alphabet. Variance is average squared deviation from mean. Mathematically, variance is calculated as:

$$\text{Var}(X) = \sum (X_i - \bar{X})^2 / N = \sum X_i^2 / N$$

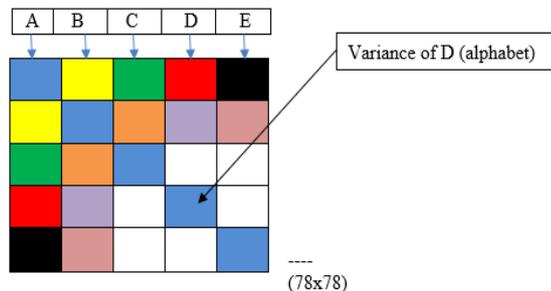


Figure 4: Representation of covariance matrix.

- b) Value in position (i, j) is equal to value in position (j, i) of the covariance matrix.

7. Calculate the eigenvalues and then with the help of eigenvalues calculate eigenvectors (Principal components). Eigenvalues are calculated for square matrix only. For example:

$$A = \begin{bmatrix} a1 & a2 \\ b1 & b1 \end{bmatrix}_{(2 \times 2)}$$

$$|A - \lambda I| = 0 \text{ where } \lambda \text{ is the eigenvalue.}$$

For a 2x2 matrix one will have 2 eigen values, for an nxn matrix 'n' eigen values are possible. For covariance matrix, 78 eigenvalues and for each eigenvalue a corresponding eigenvector are possible, that is total 78 eigenvectors are possible for covariance matrix. Example to find out the eigenvalues and eigenvectors of 2x2 matrix:

Let,

$$A = \begin{bmatrix} 5 & 4 \\ 1 & 2 \end{bmatrix}, \text{ Using the eigen value equation } |A - \lambda I| = 0$$

$$\begin{bmatrix} 5 & 4 \\ 1 & 2 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = 0$$

$$\text{Or, } \begin{vmatrix} 5 - \lambda & 4 \\ 1 & 2 - \lambda \end{vmatrix} = 0$$

$$\text{Or, } \lambda^2 - 7\lambda + 6 = 0$$

$$\text{Or, } \lambda = 6 \text{ and } 1 \text{ as eigen values.}$$

For each eigenvalues, a corresponding eigenvector is calculated as

$[A - \lambda I]X = 0$ , where X is the eigenvector where X contains 'n' (for nxn matrix) number of displacement values.

$$\begin{bmatrix} 5 & 4 \\ 1 & 2 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = 0$$

$$\begin{bmatrix} 5 - \lambda & 4 \\ 1 & 2 - \lambda \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = 0$$

For example,  $\lambda = 6$  :

$$\begin{bmatrix} -1 & 4 \\ 1 & -4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = 0$$

$$-x+4y=0; \quad x=4y; \quad \frac{x}{4} = \frac{y}{1}$$

Therefore eigenvector for eigen value  $\lambda = 6$  is (4, 1). Similarly for  $\lambda = 1$  eigenvector will be (1, -1). For covariance matrix one will get 78 eigen values and corresponding 78 eigenvectors, where each eigenvector will be of size 78x1. In MATLAB, eigen values and eigenvectors are calculated using predefined function eig(covariance matrix) which will return eigenvectors and eigen values. For example, [pc, v] = eig (covariance), where 'pc' are the eigenvectors and 'v' is the eigen values.

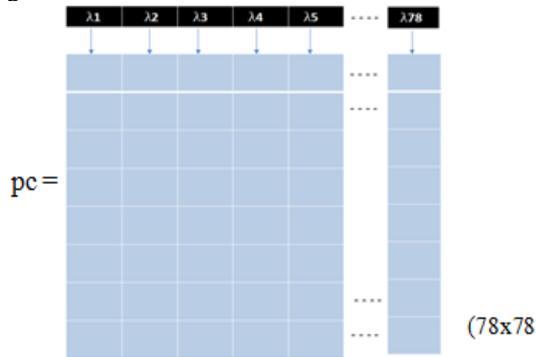


Figure 5: Eigen vectors representation in 2D form

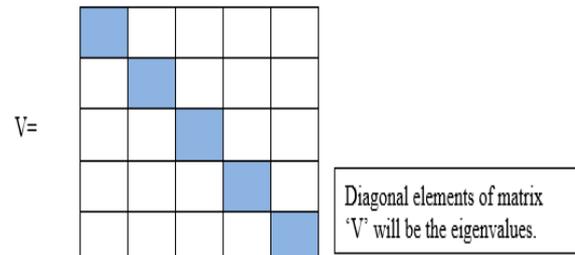


Figure 6: Eigen values Representation in 2D matrix.

8. Now sort the eigenvectors according to descending order of eigenvalues. In MATLAB one can sort it by predefined function sort (v, 'descend') which will return sorted eigenvalues (ev) and an index matrix. Index matrix is an one dimensional array which stores index values of 'v' according to descending order of eigenvalues that is  $ev=v(\text{index})$ . To sort eigenvectors (pc) use,  $pc=pc(:, \text{index})$ . It will rearrange the 'pc' according to the index matrix obtained above.
9. Convert the eigenvector of (dataset' x dataset) into eigenvector of (dataset x dataset'). Let, A be 2x4 matrix then  $A^T$  will be a 4x2 matrix.  $A^T A$  will become 2x2. Consider the eigenvectors  $V_i$  of  $A^T A$  such that:

$$A^T A X_i = \lambda_i X_i$$

The eigenvectors  $V_i$  of  $A^T A$  are  $X_1$  and  $X_2$  which are 2x1. Now multiplying the above equation with A both sides we get:

$$A A^T A X_i = A \lambda_i X_i \quad \text{or} \quad A A^T (A X_i) = \lambda_i (A X_i)$$

Eigen vectors corresponding to  $A^T A$  can now be easily calculated now with reduced dimensionality where  $A X_i$  is the Eigen vector and  $\lambda_i$  is the Eigen value.

Therefore,

$$\begin{matrix} & \swarrow & \text{pc} = \text{dataset} * \text{pc} & \searrow \\ (10000*78) & & (10000*78) & & (78*78) \end{matrix}$$

10. For better efficiency extract as many eigenvectors as required from 'pc' and store them in a 2D matrix named 'eigenvec'.
11. Reduce dataset by multiplying transpose of dataset to eigenvec.

$$\begin{matrix} \text{database} = \text{dataset}' * \text{eigenvec} \\ \downarrow \quad \downarrow \quad \downarrow \\ (78*78) \quad (78*10000) \quad (10000*78) \end{matrix}$$

12. Read an input image and do step 11 and store the result in matrix name 'ConvertedInput'. In MATLAB, [imagepath, cancelled]= imgetfile displays the open image dialog box, if one can choose an image then 'imagepath' is the full path of the file and sets the 'cancelled' value to false else cancelled value will be true. Store the input image in a 1D matrix named 'image' in column fashion, therefore size of image will be 10000x1.

$$\begin{matrix} \text{ConvertedInput} = \text{image}' * \text{eigenvec} \\ \downarrow \quad \downarrow \quad \downarrow \\ (1*78) \quad 1*10000 \quad 10000*78 \end{matrix}$$

13. Find the Euclidean distance between 'database' matrix and 'ConvertedInput' matrix.

$$\text{Distance} = \text{eqDistance}(\text{database}, \text{ConvertedInput})$$

Where, eqDistance is the user defined function to calculate the distance. Distance will be 1D matrix of size 1x78.

14. Find the index of minimum value from 'Distance' matrix.
15. Use the index value to access the name of the  $\text{index}^{\text{th}}$  row of handlist file. That name of the alphabet is the result.  $\text{Result} = \text{char}(\text{handlist}(\text{index}, 1))$ .

### III. RESULTS

The figure 7 represents the GUI made with the help of MATLAB. Input for the below interface is the image which you want to test. In this paper, database is created using three different hands of different person which indicate gestures of same alphabets that is for each alphabet there is three gestures. Test image is different from the database images.

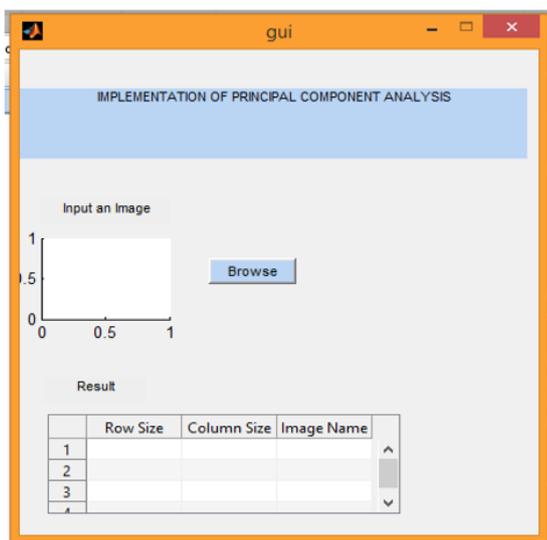


Figure 7: Front End of PCA Interface.

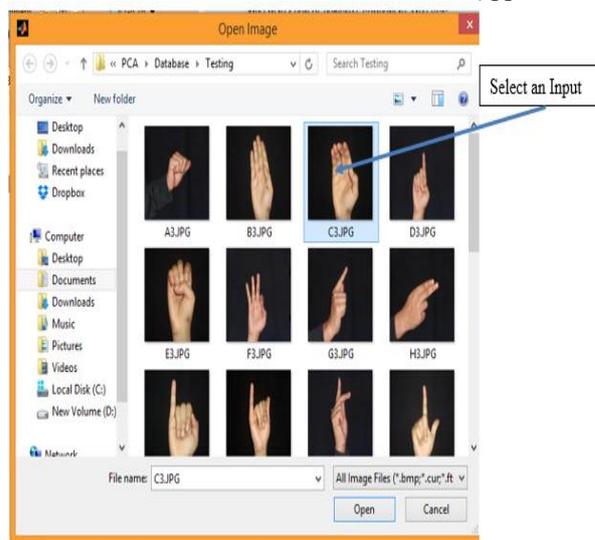


Figure 8: Database to take Input Image

In this paper, gestures are named as A1.jpg, A2.jpg, A3.jpg and A4.jpg and so on. Here A1.jpg is used to test and A2.jpg, A3.jpg, A4.jpg is used to create the database.

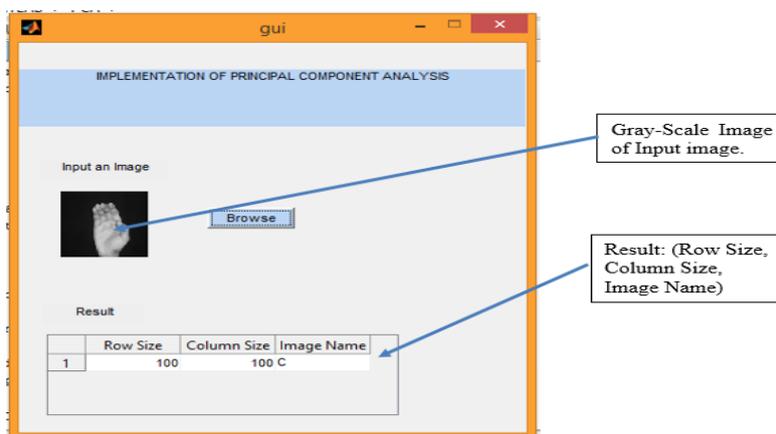


Figure 9: Output from designed System

In the back end, PCA will be implemented and the output shown will be the size of the rows and columns of the input image along with the identified value corresponding to that input image. The Euclidean distance was used to find the nearest possible match. The table shown below highlights the list of gesture representing characters that are correctly or incorrectly classified. The highlighted cells are the one which are not correctly recognized.

Table 1: List of character recognized and not-recognized by the designed system

Gesture Used to test:	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Matched Result:	A	B	C	D	E	F	G	A	C	J	K	L	M	O	O	N	Q	K	M	M	U	M	W	M	Y	Z

Out of 26 alphabets, 17 alphabets were correctly identified, which indicates 65.38% of accuracy was obtained.

#### IV. SUMMARY AND CONCLUSION

In the first phase, the understanding of PCA was well understood from dimensionality reduction point of view and its implication in image. Then, the same concept was used on hand gesture images for recognition based on Euclidean distance. The interface is capable of recognizing alphabets from hand gestures using Principal Component Analysis. Taking a dataset of 78 images (3 images for each alphabet), input images are analysed and evaluated, with an efficiency rate of 65.38% respectively. The system will take input and accordingly it displays the result, so user can show different alphabets.

However, the following are the limitation of the work which the researchers in future can focus.

- a) *Result depends on intensity of light:* The efficiency of the algorithm varies with the variation in the intensity of light while creating the database which contains different images.

- b) *Size of Hand*: Result may vary according to the size of hand.
- c) *Distance between Camera and Hand while capturing image*: if you vary the distance between Camera and Hand while capturing image for input then result may vary.
- d) *Similarity between images may lead to false result*: This paper uses ASL as standard language but there is much similarity between ‘M’ and ‘N’ which has led to incorrect results.

## REFERENCES

- [1] Mandeep Kaur Ahuja and Dr. Amardeep Singh, “Hand Gesture Recognition Using PCA”, ISSN: 2231-0711, July 2015, Vol 5, Issue 7,267-271.
- [2] Trong-Nguyen Nguyen, Huu-Hung Huynh and Jean Meunier, “Static Hand Gesture Recognition Using Principal Component Analysis Combined with Artificial Neural Network”, Journal of Automation and Control Engineering Vol. 3, No. 1, February 2015
- [3] Aditi kalsh, N.S. Grewal, Sign Language Recognition for Deaf & Dumb, ISSN: 2277 128X, Volume 3, Issue 9, September 2013
- [4] 2005, A Tutorial on Principal Component Analysis. [online] available at: [http://www.researchgate.net/publication/261404279\\_A\\_tutorial\\_on\\_princlipal\\_components\\_Analysis](http://www.researchgate.net/publication/261404279_A_tutorial_on_princlipal_components_Analysis) [Accessed 5 Nov 2015]
- [5] Jonathon Shlens, “A Tutorial on Principal Component Analysis”, Systems Neurobiology Laboratory, Salk Insitute for Biological Studies La Jolla, CA 92037 and Institute for Nonlinear Science, University of California, San Diego La Jolla, CA 92093-0402 (Dated: December 10, 2005; Version 2)
- [6] K. Imagawa, H. Matsuo, R. Taniguchi, D. Arita and S. Igi “Recognition of local features for camera based sign language recognition system”, Proc. International conference on pattern recognition, volume 4, pp.849-853 2000
- [7] Imran S. Bajwa, M. Shahid Naweed , M. Nadim Asif , S. Irfan Hyder, “Feature Based Image Classification by using Principal Component Analysis”, CIST-Journal of Graphics, Vision and Image Processing.