

## MATHEMATICAL MODELING TO ENHANCE SECURITY FEATURE FOR PASSWORD AUTHENTICATION USING MATLAB

VARUN KUMAR<sup>a</sup>, P.N. PATHAK<sup>b1</sup>, MANU PRATAP SINGH<sup>c</sup> AND M.K. GUPTA<sup>d</sup>

<sup>a</sup>Department of Mathematics, Vidya College of Engineering Meerut, India

<sup>b</sup>Department of Mathematics, Axis College, Kanpur, India

<sup>c</sup>Department of Computer Science, IBS, Khandari, Agra, India

<sup>d</sup>Department of Mathematics CCS University, Meerut, India

### ABSTRACT

Since the beginning of computer science, information technology or to be particular image processing to enhance security system in cyber world the mathematics is working as backbone of all. Image processing consists of a wide variety of techniques and mathematical tools to process an input image. The Principal Component Analysis (PCA) is one of the most successful mathematical techniques that have been used to compress an image. Further various edge deduction tools or neural network may be used for password authentication. The most common computer authentication method is to use alphanumerical usernames and passwords. This method has been shown to have significant drawbacks. In this paper, we proposed a new scheme to enhance security issues using PCA as an image compression and authentication tool and MATLAB to perform the typical mathematical operations.

**KEYWORDS:** Image Recognition, Image Compression, Principal Component Analysis, MATLAB

Alphanumeric passwords are the most commonly used way of authenticating users in computer systems. One of the disadvantages of alphanumeric passwords is that they are hard to remember. Patrick, *et al.* (2003) point out that there are three major areas where human computer interaction is important: authentication, security operations, and developing secure systems. Here we focus on the authentication problem and try to develop a new method to use images as a password. An image is processed as soon as we start extracting data from it. The data of interest in object recognition systems are those related to the object under investigation. An image usually goes through some enhancement steps, in order to improve the extractability of interesting data and subside other data. Lots of mathematical tools and software based on mathematics are behind all these (Belhumeur *et al.*, 1997) (Swets and Weng, 1999) (Website, 2000). Over the past few years, several face recognition systems have been proposed and extensive research has been carried out in the area of image processing based on principal components analysis (Kirby, 2000) (Moon and Phillips, 1998). PCA is a statistical method under the broad title of factor analysis. The purpose of PCA is to reduce the large dimensionality of the data space (observed variables) to the smaller intrinsic dimensionality of feature space (independent variables), which are needed to describe the data economically (Elwards, 1991) (Nayar *et al.*, 1996). This is the case when there is a strong correlation between observed variables. The jobs which PCA can do are prediction, redundancy removal, feature extraction, data compression, etc. Because PCA is a classical technique

which can do something in the linear domain, applications having linear models are suitable, such as signal processing, image processing, system and control theory, communications, etc. Face recognition has many applicable areas (Dietterich and Bakiri, 1995) (Kirby and Sirovich, 1990) (Wiedenbeck *et al.*, 2005). Moreover, it can be categorized into face identification, face classification, or sex determination. Principal component analysis is the general name for a technique which uses sophisticated underlying mathematical principles to transform a number of possibly correlated variables into a smaller number of variables called principal components. The main idea of using PCA for face recognition is to express the large 1-D vector of pixels constructed from 2-D facial image into the compact principal components of the feature space. PCA uses a vector space transform to reduce the dimensionality of large data sets. Using mathematical projection, the original data set, which may have involved many variables, can often be interpreted in just a few variables (the principal components). It is therefore often the case that an examination of the reduced dimension data set will allow the user to spot trends, patterns and outliers in the data, far more easily than would have been possible without performing the principal component analysis (Sirovich and Kirby, 1987) (Swets and Weng, 1996). We will also use singular value decomposition method to perform PCA. There are lots of mathematics working behind the PCA we will take a quick look on it.

<sup>1</sup>Corresponding author

**MATHEMATICAL TOOLS REQUIRED FOR PCA**

MATLAB considers grayscale images as 'objects' consisting of two components, a matrix of pixels, and a colourmap. If an image has 256X256 number of pixels then the image is stored in a 256X256 matrix). The colourmap is a 256× 3 matrix. For RGB colour images, each image can be stored as a single 256×256×3 matrix, where the third dimension stores three numbers in the range [0, 1] corresponding to each pixel in the 256X256 matrix For a grayscale image such as the one we are dealing with, the colourmap matrix has three identical columns with a scale representing intensity on the one dimensional grey scale. Each element of the pixel matrix contains a number representing a certain intensity of grey scale for an individual pixel. MATLAB displays all of the 256X256 pixels simultaneously with the correct intensity and the grayscale image that we see is produced. The 256X256 matrix containing the pixel information is our data matrix, X. We will perform a principal component analysis of this matrix, using the singular value decomposition method. The steps involved are exactly as described above and summarized in the following MATLAB code. To understand mathematical concept we will go through the following steps;

- Step 1. To extract the information from the image and acquire data and find Mean, variance & Standard deviation the data
- Step 2. To construct the covariance matrix from the acquired data
- Step 3. To find eigen values and eigen vectors the acquired data
- Step 4. Short eigen vectors and find principle eigen vectors the acquired data
- Step 5. Image compression using these principle eigen vectors

As we see that many of the terms are belongs to statistics and linear algebra, which can't be covered in a research paper. But to understand the topics in brief we are we go through the few definitions. But the readers are advised to go through the books (Devore and Peck, 1997) (David, 2000) to understand the topic deeply. Throughout the paper we will consider the symbol X to refer to this entire set of numbers and to denote an individual number in this data set and we will use subscripts on the symbol X. Let  $X_1, X_2, \dots, X_n$  denote the random variables for a sample of size n.

The mean of the data set is a random variable defined by

$$Mean(\bar{X}) = \frac{1}{n} \sum_{i=1}^n X_i \tag{1.1}$$

In case of frequencies of the above variable are  $F_1, F_2, \dots, F_n$  respectively then Mean

$$Mean = \frac{\sum_{i=1}^n F_i X_i}{\sum_{i=1}^n F_i} \tag{1.2}$$

The Standard Deviation (SD) of a data set is a measure of how spread out the data is. As a formula

$$SD = \sqrt{\frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})^2} \tag{1.3}$$

In case of frequencies of the above variable are  $F_1, F_2, \dots, F_n$  respectively then standard deviation. The square of the standard deviation variance ( $\sigma$ ) which is another measure of the spread of data in a data set. In fact it is almost identical to the standard deviation. In everyday life we observe that there exist certain relationship between pairs of variables Covariance is always measured between 2 dimensions. The formula

$$cov(X, Y) = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{n} \tag{1.5}$$

**The Covariance Matrix**

As covariance is always measured between two dimensions. If we have an 3-dimensional data set, then to make up the covariance matrix for an imaginary 3 dimensional data set, using the usual dimensions  $\langle x, y \& z \rangle$ . Then, the covariance matrix has 3 rows and 3 columns, and the values are this:

$$\sigma = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{bmatrix} \tag{1.6}$$

And for n dimensional data set, we can calculate

$$C_2^n = \frac{n!}{(n-2)!*2!}$$

**Eigen values & Eigen Vectors**

If A is a  $n \times n$  matrix, then  $[X] \neq \bar{0}$  is an eigenvector of A if

$$[A][X] = \lambda[X] \quad (1.7)$$

where  $\lambda$  is a scalar and  $[X] \neq 0$ . The scalar  $\lambda$  is called the eigenvalue of  $A$  and  $[X]$  is called the eigenvector corresponding to the eigenvalue  $\lambda$ .

### Singular Value Decomposition

Suppose  $M$  is an  $m \times n$  matrix whose entries come from the field  $K$ , which is either the field of real numbers or the field of complex numbers. Then there exists a factorization of the form  $M = U\Sigma V^*$  where  $U$  is an  $m \times m$  unitary matrix over  $K$ , the matrix  $\Sigma$  is an  $m \times n$  diagonal matrix with nonnegative real numbers on the diagonal, and the  $n \times n$  unitary matrix  $V^*$  denotes the conjugate transpose of  $V$ . Such a factorization is called the singular value decomposition of  $M$ . The diagonal entries  $\sigma_i$  of  $\Sigma$  are known as the singular values of  $M$ . A common convention is to list the singular values in descending order. In this case, the diagonal matrix  $\Sigma$  is uniquely determined by  $M$ .

### Proposed Scheme

The most common computer authentication method is for a user to submit a user name and a text password. One of the main problems is the difficulty of remembering passwords. Studies have shown that users tend to pick short passwords or passwords that are easy to remember (Belhumeur *et al.*, 1997). Unfortunately, these passwords can also be easily guessed or broken. So using the tools and MATLAB we are proposing a new scheme to enhance the security, and is useful where high security is required. I will thanks to MATLAB to help me in very long calculations which are required in the research work. When a new user doing sign me up in any website or any network which is secured with password, user can see there are two windows. One is for entering login ID and other is for entering password.

We are proposing a new unique scheme to enter Login ID as well as password. An image will be associated with each a to z alphabets. Also there is a hidden chain of elementary transformations (ETs) (row & column) / rotation is defined. User is free to choose his own image as well as he is free to choose the already stored chain of elementary transformations (row & column) or he can create a new chain of transformations. Once the unique login ID is finalized the user is supposed to choose the password. When the user will click on password window, a new window will be open which contains 25 (5X5) images, user is required to choose at least 2(or more but even number of) images out of 25 so

that a new image has been generated using these 2 images(say PASS-IM ie PASSword IMages). To create the PASS-IM first we convert the image in same size then use matlab operation 'imadd' to find overlap images or to combine the images using the given MATLAB commands. Further to increase more security, number of images in the set of images may be increased by associating more images with 0 to 9 numerical digits and with each special character available in standard key board.

Now the mathematical work started, This PASS-IM will be changed to RGB Image then Gray level image using MATLAB. Then MATLAB will be converted it to a covariance matrix using gray level value or pixel value. Then this matrix is transformed using the chain of elementary transformations (row & column) which has already been selected by the user. Now eigen values and eigen vectors have been found by using MATLAB. Here is where the notion of data compression and reduced dimensionality comes into it. The eigenvector with the highest eigen value is the principle component of the data set. In general, once eigenvectors are found from the given matrix corresponding to each eigen vectors, the next step is to order them by eigen value, highest to lowest. This gives us the components in order of significance. Now, we can decide to ignore the components of lesser significance. We do lose some information, but if the eigen values are small, we don't lose much information. If we leave out some components, the final data set will have fewer dimensions than the original. To be precise, if we originally have  $n$  dimensions in our data, and so we calculate first  $p$  eigenvectors and eigen values, and then we choose only the first  $p$  eigenvectors, then the final data set has only  $p$  dimensions. Then a new matrix will be created as eigen vectors in column form is called MM (Modal Matrix) and will stored in our data base for authentication purpose.

Now when the next time the same user will try to login he has to select the same sequence of image which is stored as unique Login ID then he has to select the required chain of password from the given images to construct the PASS-IM. When the PASS-IM is generated the MATLAB program will create a covariance matrix as said above, then the sequence of elementary transformations (row & column) already attached with every image in Login ID will be transformed it. Then eigen values and eigen vectors will be calculated by MATLAB and select principle component of the data set. Then a matrix will be generated using this principle eigen

vectors and ready to compare with stored MM by term by term matching.

### Image Compression in MATLAB

Let us start with a data set that is represented in terms of an  $m \times n$  matrix,  $X$  where the  $n$  columns and the  $m$  rows are the samples (e.g. pixel values or gray level values)

$$X = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ x_{m1} & x_{m2} & \dots & x_{mn} \end{bmatrix}$$

Since we have a row vector for each variable  $X_1, X_2, \dots, X_m$  with  $X_i = [x_{i1} \ x_{i2} \ \dots \ x_{in}]$ . Each of these vectors contains all the samples for one particular variable. So for example,  $x_i$  is a vector of the  $n$  samples for the  $i^{\text{th}}$  variable. It therefore makes sense to consider the following matrix product.

$$C_x = \frac{X_i X_i^T}{n-1} \quad \forall i = 1, 2, \dots, m \quad (1.8)$$

If we look closely at the entries of this matrix, we see that we have computed all the possible covariance pairs between the  $m$  variables. Indeed, on the diagonal entries, we have the variances and on the non diagonal entries, we have the covariance. This matrix is therefore known as the Covariance Matrix. Now let us transform the  $C_x$  using a chain of elementary transformation by elementary matrix and then extract eigen values and vectors from it. Find a new square orthogonal matrix  $P$  (MM) by selecting principle component from  $C_x$  using MATLAB. Let us linearly transforming the original data matrix using the relation  $Y = PX$ , for orthogonal matrix  $P$ . The matrix  $Y$  will appear as compress image from the

previous one depends how many principle component we have chosen. Returning to the original  $m \times n$  data matrix,  $X$ , let us define a new  $n \times m$  matrix,  $Z$ :



$$Z = \frac{1}{\sqrt{n-1}} X^T \quad (1.9)$$

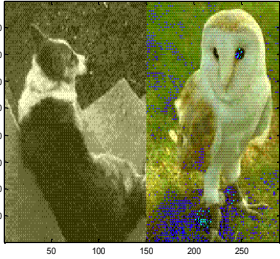

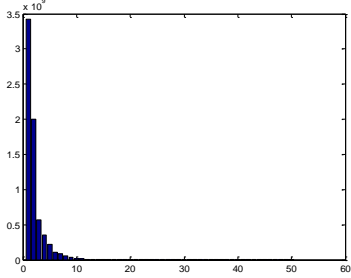

Recall that since the  $m$  rows of  $X$  contained the  $n$  data samples, we subtracted the row average from each entry to ensure zero mean across the rows. Thus, the new matrix,  $Z$  has columns with zero mean. Consider forming the  $m \times m$  matrix,  $Z^T Z = C_x$ . From the discussion in the previous section, the principal components of  $X$  are the eigenvectors of  $C_x$ . Therefore, if we perform a singular value decomposition of the matrix  $Z^T Z$ , the principal components will be the columns of the orthogonal matrix  $C_x$ . To project the original data onto the directions described by the principal components we use the relation  $X = P^T Y$  as  $P$  is orthogonal matrix  $P^{-1}$  is  $P^T$ .

## EXPERIMENT AND RESULTS

### Part1: Image Compression

During the subsequent analysis, we shall work with lots of images for example a user chosen two images out of 25 images Figure 1 & 2 ( dog.gif, owl.gif) during the image processing and image compression. Then the combination of these two pictures is considered and displayed in Figure 3: two images.gif. We will use it in MATLAB to perform the following analysis, though the principles can be applied in other computational packages. To demonstrate the compression we have chosen the first 10 (Figure 5), 40 (Figure 6) principal components. Further for the experiment purpose, the images can be converted to the Java class java.awt. image (figure 6) from MATLAB workspace, by using the command 'im2java2d'.

<pre> &gt;&gt;[A,map]=imread('two images.gif'); &gt;&gt;A=double (A); &gt;&gt;image (A),colormap (map); &gt;&gt;axis off, axis equal &gt;&gt;&gt;&gt;[m n]=size (A) &gt;&gt;mn = mean (A,2); &gt;&gt;X=A-repmat(mn,1,n); &gt;&gt;Z=1/sqrt(n-1)*X'; &gt;&gt;covZ=Z'*Z; &gt;&gt;eig ('covZ') &gt;&gt;[eigvec, eigval]=eig(covZ); &gt;&gt;sort ('eigvec'); &gt;&gt;P=rot90('covZ'); &gt;&gt;[U,S,V]=svd(covZ);                 </pre>	<div style="text-align: center;">  <p>Figure1: dog.gif</p> </div> <hr/> <div style="text-align: center;">  <p>Figure2 : dog.gif</p> </div>
--	--

<pre> &gt;&gt;variances=diag (S).*diag(S); &gt;&gt;bar (variances (1:50)); &gt;&gt;PCs=10 &gt;&gt;VV=V(:,1:PCs); &gt;&gt;P=V(:,1:PCs); &gt;&gt;Y=P*X; &gt;&gt;ratio=256/(2*PCs+1); &gt;&gt;MM=P*Y &gt;&gt;MM=MM+repmat(mn,1,n); &gt;&gt;image(MM),colormap(map),axis off; &gt;&gt;PCs=40; &gt;&gt;P=V(:,1:PCs); &gt;&gt;Y=P*X; &gt;&gt;ratio=256/(2*PCs+1); &gt;&gt;MM=P*Y &gt;&gt;MM=MM+repmat(mn,1,n); &gt;&gt;image(MM),colormap(map),axis off;         </pre>	 <p>Figure3 : two images.gif</p>
 <p>Figure 5: with 10 PCs</p>	 <p>Figure 4: first 50 Variances</p>  <p>Figure 6: with 40 PCs</p>

**Part 2: Password Authentication Using MATLAB**

There are many mathematical tools are available for image authentication (Devijver and Kittler, 1982). The MATLAB has various features which can be utilized for authentication purpose. There are various edge detection algorithm (Maini and Aggarwal, 2006) (Gao and Leung, 2002), which can be used to detect the edges as well as image matching purpose. The following code are suggested in MATLAB as an example to deduct edges from the image further match the image and show the accuracy of the result. If comparison will be successful user will be login.

```

>>[x1,y1,z1] = size(image1);
>>if(z1==1);
>>else
>>image1 = rgb2gray(image1);
>>[x2,y2,z2] = size(image2);
>>if(z2==1);
        
```

```

>>else
>>image2 = rgb2gray(image2);
>>edgeimage1 = edge(image1,'prewitt'); % display the image(figure6:13)
>>edgeimage2 = edge(image2,'prewitt'); % display the image(figure6:14)
% further the image can be matched using the following command
>>A = imread('image1.jpg');
>>B = imread('image2.jpg');
>>C = corr2(A,B); % finding the correlation between two images
>>if c==1
disp('The images are same') %output display
else
disp('the images are not same')
end;
        
```

## CONCLUSION

The alphanumeric passwords have some disadvantages that biometrics challenge to deal with it but is expensive. The key purpose to choose graphical passwords is that pictures are easier to remember than text sequences. Many researchers have put efforts to implement it (Patrick *et al.*, 2003) (Wiedenbeck *et al.*, 2005). In the paper we are exploring this area. Pictures are attractive and this increases the memorability of the passwords. This work proposes an alternative scheme for picture passwords authentication that involve various mathematical tools including PCA and MATLAB. The scheme is user friendly and increases the demanded security in cyber world. However, further work can involve a questionnaire study designed to determine the usability of the scheme. Further work should also be done to determine different kinds of attacks against the proposed scheme.

## REFERENCES

- Belhumeur P., Hespanha J. and Kriegman D., 1997. "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection". IEEE Transactions on Pattern Analysis and Machine Intelligence, **19**(7):771-720.
- David L., 2000. "Linear Algebra and It's Applications". Addison-Wesley, New York.
- Devore J. and Peck R., 1997. "Statistics: The Exploration and Analysis of Data", Thrid Edition. Brooks Cole.
- Devijver P.A. and Kittler J., 1982. Pattern Recognition: A Statistical Approach, Englewood CliIs, London.
- Dietterich T. and Bakiri G., 1995. "Solving multiclass learning problems via error-correction output code". Journal of Artificial Intelligence Research, **2**: 263-286.
- Elwards J., 1991. "A User guide to Principle Component". Wiley Series in Probablity and Mathematical Statistics.
- Gao Y. and Leung K.H., 2002. "Study and Comparison of various image edge detection techniques" IEEE transactions on pattern analysis and machine intelligence, **24**(6).  
<http://fonsg3.let.uva.nl:8001/service/statistics.html>. Website, 2000.
- Kirby M. and Sirovich L., 1990. "Application of the Karhunen-Loeve Procedure for the Characterization of Human Faces". IEEE Trans. on Pattern Analysis and Machine Intelligence, **12**(1): 103-107.
- Kirby M., 2000. "Dimensionality Reduction and Pattern Analysis: an Empirical Approach". Wiley (in press).
- Maini R. and Aggarwal H., 2006. International Journal of Image Processing (IJIP), **3**(1).
- Moon H. and Phillips J., 1998. "Analysis of pca-based face recognition algorithms." Empirical Evaluation Techniques in Computer Vision. IEEE Computer Society Press.
- Nayar K., Nene A.S. and Murase H., 1996. "Real-Time 100 Object Recognition System". In Proceedings of ARPA Image Understanding Workshop. Morgan Kaufmann.
- Patrick S., Long A.C. and Flinn S., 2003. "HCI and Security Systems," presented at CHI, Extended Abstracts (Workshops). Ft. Lauderdale, Florida, USA.
- Sirovich L. and Kirby M., 1987. "A low-dimensional procedure for the characterization of human faces". The Journal of the Optical Society of America, **4**: 519-524.
- Swets D. and Weng J., 1996. "Using discriminant eigenfeatures for image retrieval". IEEE Transactions on Pattern Analysis and Machine Intelligence, **18**(8): 831-836.
- Swets D. and Weng J., 1999. "Hierarchical discriminant analysis for image retrieval". IEEE Transactions on Pattern Analysis and Machine Intelligence, **21**(5):386-401.
- Wiedenbeck S., Waters J., Birget J.C., Brodskiy A. and Memon N., 2005. "Authentication using graphical passwords: Effects of tolerance and image choice," in Symposium on Usable Privacy and Security (SOUPS). Carnegie-Mellon University, Pittsburgh.