Effect of Chinese Remainder Theorem on Principal Component Analysis

MADANDOLA Tajudeen Niyi [1], GBOLAGADE Kazeem Alagbe [2], YUSUF-ASAJU Ayisat Wuraola [3]

[1]*(Department of Computer Science / Kwara State College of Education, Oro, Nigeria)*

[2]*(Department of Computer Science / Kwara State University, Malete, Nigeria)*
[3]*(Department of Computer Science / University of Ilorin, Nigeria)*

Abstract— Principal Component Analysis (PCA) has proved to be one of the most successful dimensionality reduction algorithms but its computational time and memory usage requires improvement. As a result of this, Chinese Remainder Theorem (CRT) is employed. YALE face database which contains frontal gray scale face images of 15 people, with 11 face images of each subject, giving a total of 165 images is adopted. 120 images are use for training while 45 images are use for testing. The performance metrics to determine the effect of CRT on PCA in terms of computational time and recognition accuracy are recognized index in database, training time and testing time. In the experiment, the average training time and average testing time for PCA without CRT are 28.5026 seconds and 1.8146 seconds respectively while the average training time and average testing time for PCA with CRT are 26.6393 seconds and 1.6863 seconds respectively. Also, it is observe that out of the 45 images use for testing, 32 images are recognise and 13 images do not match when employing CRT with PCA for face recognition and the same result is reveal when CRT is not employ to PCA. Column chart is use to show the relationship between Training time and average testing time for PCA with and without CRT. The research reveals that employment of CRT to PCA improves its computational time and memory usage by reducing its training and testing time but does not have any effect on its recognition accuracy.

Index Terms— Chinese Remainder Theorem, Computational Time, Dimensionality reduction, Yale Database and Principal Component Analysis..

1 INTRODUCTION

The world is in an era of security risks and challenges, as a result of that, several techniques have been developed for both identification and verification. Human face as a key to security (face recognition technology) has received laudable acceptance by both law enforcement and other agencies. Major benefits of facial recognition are that it is non-instrustive, hands-free, continuous and accepted by most users [1]. The most popular face recognition literature nowadays uses appearance-based method. This is so due to the fact that the method does not need creation of models for objects because each object model has been intelligently described by the selected sample images of the objects. However, Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA, also known as Fisher Discriminant Analysis - FDA) are two main techniques used for data reduction and feature extraction in the appearance-based approaches. The two techniques have been proved to be very successful. LDA algorithm selects features that are most effective for class separability while PCA selects features important for class representation [2]. It has been argued by [3] that when the training dataset is small, PCA can outperform LDA, and also that PCA is less sensitive to different training datasets. PCA and LDA are the two most popular techniques usually used for dimensionality reduction [4]. [5] were among the first to apply PCA to face images, and found that it effectively and efficiently represents pictures of faces into its eigenface components. Similarly, it was also disclosed that PCA is an optimal compression scheme that minimizes the mean squared error between the original images and their reconstructions for any given level of compression [5][6]. Unfortunately, PCA is a good face feature extractor for face recognition but its computational time and memory usage requires improvement. In view of that, this study introduces Residue Number System (RNS) to PCA in order to reduce its computational time. Chinese Remainder Theorem (CRT) which is one of the methods of RNS is desirable because the computation can be parallelized while Mixed Radix Conversion (MRC) is by its very nature a sequential process and requires a large number of arithmetic operations. The dataset in Yale face database is employed for testing and training the system. Recognition index, Training time and Testing time were used as the performance metric for the study.

2 LITERATURE REVIEW

2.1 Principal Component Analysis

The central idea of exploiting PCA for face recognition is to express the bulky 1-D vector of pixels created from 2-D facial image into the miniature principal components of the feature space. This can be called eigenspace projection [7].

PCA is a powerful statistical tool for face recognition system. It is one of the reliable algorithms for dimensionality reduction. The PCA is to diminish the huge dimensionality of the data space (observed variables) to the smaller intrinsic dimensionality of feature space (independent variables), which is needed to describe the data economically [8]. The opening principal component is the linear combination of the original dimensions that has the highest variability. The last principal component is the linear combination with the maximum variability, being orthogonal to the n-1 first principal components. PCA is a common technique for finding patterns in data, and expressing the data as eigenvector to highlight the similarities and differences between different data set.

The following steps summarize the PCA process:

1. Let {S1, S2, …, SM} be the training data set. The average (Arg) is defined by:

$$Arg = \frac{1}{M}\sum_{i=1}^{M} Si \qquad \text{……….. (1)}$$

2. Individual element in the training data set differs from by the vector $Ei=Si-Arg$. The covariance matrix obtained as:

$$Kov = \frac{1}{M}\sum_{i=1}^{M} Ei.Ei^{T} \qquad \text{……….. (2)}$$

3. Select M′ significant eigenvectors and compute the weight vectors Zik the training data set, where k varies from 1 to M′.

$$Z_{ik} = U^{T}_{k}.(S_i - Arg), \forall\, i, k \qquad \text{……….. (3)}$$

2.2 Residue Number System

There are numerous advantages of Residue Number System (RNS) over the conventional binary systems. This is as a result of the inherent properties of RNS which include carry free operations, parallelism, modularity and fault tolerance. As a result of these properties, RNS would be required for performing parallel binary addition in a computer of equal component operating speed and number range. However, in RNS, the base consists of an N-tuple of integers, $\{m_i\}_{i=1}^{N}$ where individual member is called a modulus as shown in Fig. 1. Given any base, the RNS representation,$\{r_i\}_{i=1}^{N}$ where r_i are integers defined by a set of N equations

$x = q_i m_i + r_i$ (*) i = 1, 2, ..., N and q_i is an integer so chosen that $0 \le r_i < m_i$.

It is clear that $q_i$is an integer value of the quotient x/m_i which is denoted by $[x/m_i]$.

The quantity r_i is the least positive integer remainder of the division of x by m_i and is denoted as $x \bmod m_i$,

i.e., $|x|_{m_i}$. Eqn (*) above can be re-written as;

$$x = m_i[x/m_i] + |x|_{m_i}$$

The existing reverse converters are stemming from either the Mixed Radix Conversion (MRC) or the Chinese Reminders Theorem (CRT). CRT is desirable because the computation can be parallelized while MRC is by its very nature a sequential process and requires a large number of arithmetic operations [9]. Due to the merits of CRT, it will be adopted in study.

According to [11] the traditional CRT is defined as follows: for a moduli set {m_1, m_2, m_3, …, m_k} with the dynamic range M = $\prod_{i=1}^{k} m_i$, the residue number (x_1, x_2,x_3, …, x_k) can be changed into the decimal number X,
as follows:

$$X = \left|\sum_{i=1}^{k} M_i \left|M_i^{-1} x_i\right|_{m_i}\right|_{M},$$

Where M =$\prod_{i=1}^{k} m_i$, $M_i = \frac{M}{m_i}$, and M_i^{-1} is the multiplicative inverse of M_i with respect to m_i.

The CRT is useful in reverse conversion as well as several other operations. Conversion from residue numbers to conventional equivalents seems relatively straightforward on the basis of the Chinese Remainder Theorem (CRT).

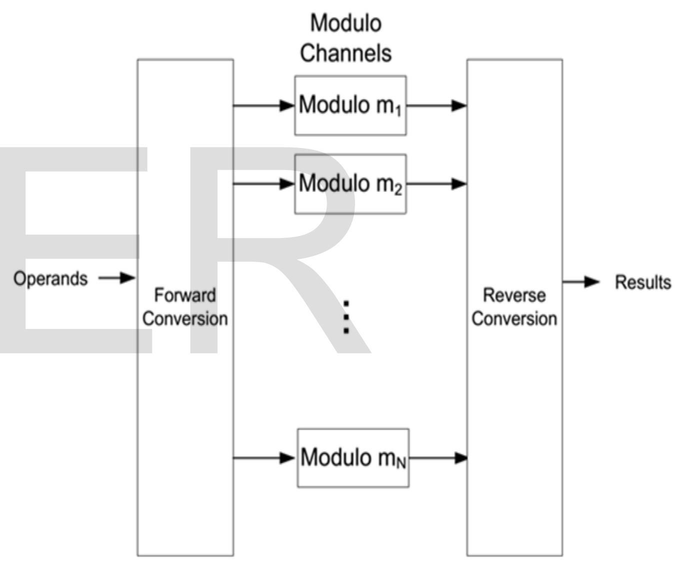

Fig. 1: General structure of an RNS processor [10]

2.3 Related Literature

[12] developed Dimensionality Reduction optimizer simply tagged DROP. The study manipulates PCA based dimensionality reduction to recognize and return a low-dimensional representation of the input. The system implements PCA on a small sample to obtain a candidate transformation, then increases the number of samples until termination. Greedy heuristic was employed to estimate the optimal stopping point. It has speedups of up to 5× against Singular-Value Decomposition (SVD)-based PCA. Its efficiency is ascertained by the dataset's variety. [13] improves the Performance and Accuracy of Local PCA (LPCA). A novel SortCluster LPCA (SC-LPCA) was proposed. SortCluster LPCA was compared with the

ISSN 2229-5518

original LPCA for compression of Patten Recognition Technique (PRT) and Bidirectional texture function (BTF) datasets. SortCluster LPCA algorithm significantly reduces the cost of the point-cluster classification stage. In addition, Adaptive Modified PCA (AMPCA) for Face Recognition" combined Sanger's adaptive algorithm for computation of effective eigenvectors with QR decomposition algorithm for adaptive estimation of related eigenvalues was examined by [2]. An on-line face recognition system was constructed and trained it with a sequence of input images from YALE face database. Euclidean distance was used as a classifier for incoming test data. The result obtained revealed that as the dimensionality of sub-space increase, the error rate decrease and as a result recognition rate improved. It was also deduced that normalization of feature vectors improves performance of classifier.

3 METHODOLOGY

i. Image acquisition through Yale database
ii. Design and Implement a system that employ CRT with PCA
iii. Comparison of performance metrics (Training time, testing time, recognition accuracy) PCA with and without CRT Using column chart to compare results

3.1 Database

The database which contains frontal gray scale face images of 15 people, with 11 face images of each subject, giving a total of 165 images as shown *in* Fig. 2. The images were from different lighting variations (left-light, center-light, and right-light), with and without spectacle and under different facial expression variations (normal, happy, sad, sleepy, surprised, and wink). 120 images were used for training while 45 were used for testing.

Fig. 2: Images used for training the database [14]

TABLE 1: Analysis of the Data used for the used

VARIABLES	FREQUENCY
Number of persons	15
Number of sample per persons	11
Number of Total sample	165
Number of Training set	120
Number of Testing sample	45

3.2 System Design

In this study MATLAB R2015a was used to implement effect of PCA with and without CRT on Intel(R) Celeron (R) CPU with 1.60GHz Processor speed. The experiment was conducted with total of 165 facial images, detailed *in* TABLE 1. Recognized index in Database, Training time and Testing time were used as performance metrics to determine the effect of CRT on PCA in terms of computational time and recognition accuracy. The system consists of number of modules: image acquisition, feature extraction and recognition accuracy as shown in Fig. 3.

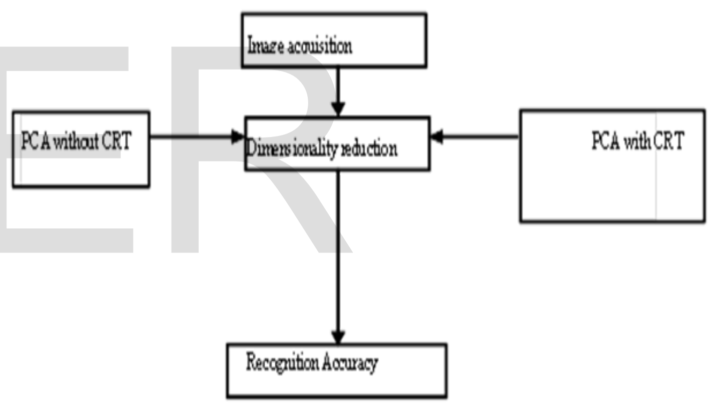

Fig. 3: Research Outline

3.3 Implementation of Chinese Remainder Theorem with PCA

In order to study dependencies between variables one employ covariance matrix $\sum$
If :
Matrix K = Warehoused several dimensional data
$\sum$ = Covariance matrix, Each row = a *sample,* Each column = a *variable*
Two variables correlated if there is a linear relationship between them
Having a covariance matrix $\sum$, then occur Eigenvector matrix (v) and Eigenvalue matrix referred to as diagonal Matrix (Λ).
1. PCA normalizes the data matrix K to Zero mean and multiply by matrix P

ISSN 2229-5518

PCA(K) = $(K - \mu_K)P$

- Resulted to Vector Subspace;

Z = PCA(K) = $(K - \mu_K)P$; Then

2. calculate covariance matrix X

$$\textstyle\sum_K = (K - \mu_K)' (K - \mu_K)$$
$$\textstyle\sum_z = [(K - \mu_K)p]' [(K - \mu_K)p]$$
$$\textstyle\sum_z = p' (K - \mu_K)' (K - \mu_K)P$$
$$\textstyle\sum_z = P' \sum_K P$$

- If P choose to be eigenvector matrix v, then

$$\textstyle\sum_z = v' \sum_K v$$

3. find [orthonormal] eigenvectors of Σ

$$\textstyle\sum_K v = v\Lambda$$
$$\textstyle v' \sum_K v = v^T v\Lambda$$
$$\textstyle\sum_z = \Lambda$$

Pass the Eigenvalue gotten from PCA to Chinese Remainder Theorem then follow the below steps:

Step 1: Find the product of all the numbers in the first array.

```
for(int a=0; a<digit.leng; a++ ){
  prod *= digit[a];
}
```

Step 2: Find the partial product of each number.

Partial product of *n*= product/*n*

```
for(int a=0; a<num.leng; a++){
  partialProduct[i] = prod/digit[a];
}
```

Step 3. Find the modular multiplicative inverse of digitr[a] modulo partial Prod[a].

Here find the inverse using the extended Euclidean algorithm

```
public static int computeInverse(int k, int m){
     int y = m, s, p;
     int c = 0, d = 1;
        if (m == 1)
       return 0;
        // Apply extended Euclid Algorithm
     while (k > 1)
     {
       // p is quotient
       p = k / m;
       s = m;
           // now proceed same as Euclid's algorithm
       m = k % m;
            k = s;
       s = c;
         c = d - p * c;
         d = s;
     }
         // Make c1 positive
     if (d < 0)
      d += y;
         return d;
   }
```

Step 4: Final Sum

sum += partialProduct[a] * inverse[a] * rem[a];

Step 5: Return the smallest C

In order to find the smallest of all solutions, perform division on the summation from Step 4 by the product from step 2.

return sum % product;

Thus, found C.

The smallest out of the Eigen space vector is the new feature called C. It will then be passed for Training.

4 RESULT AND DISCUSSION

Critical study of TABLE 2 and Fig. 6 & 7 show the computational time (Training and Testing time) with CRT and without CRT on PCA using the same images and the same system requirements. It was revealed that the Traning Time and Testing Time on each image without CRT is more than the Traning Time and Testing Time on the same images with CRT. For example, Training Time and Testing Time on image 1 when using PCA without CRT are 27.4796 seconds and 1.6871 seconds respectively while the Training Time and Testing Time for the same image 1 when using PCA with CRT are 26.8580 seconds and 1.6478 seconds respectively, as shown in TABLE 2 and Fig. 4 & 5.

TABLE 2: Analysis of Computational Time for both PCA with and without CRT

IMAGE	PCA TRAINING TIME (Seconds) WITHOUT CRT	PCA TRAINING TIME (Seconds) WITH CRT	PCA TESTING TIME (Seconds) WITHOUT CRT	PCA TESTING TIME (Seconds) WITH CRT
1	27.4796	26.8580	1.6871	1.6478
2	29.8482	25.9308	1.9101	1.6449
3	28.9859	26.0907	1.8258	1.6362
4	29.4481	25.9391	1.8184	1.6207
5	29.1704	26.3662	1.7808	1.6371
6	28.9506	25.9642	1.8843	1.6574
7	28.9350	27.0066	1.8334	1.6834
8	29.4120	25.9135	1.8024	1.6871
9	29.1765	26.0492	1.8808	1.6325
10	29.3168	26.0061	1.8096	1.6658
11	29.4082	26.0519	1.8453	1.6428
12	29.2889	25.9426	1.8336	1.6494
13	29.4092	26.0867	1.8151	1.7136
14	28.0127	26.2332	1.8186	1.6464
15	27.9182	26.2384	2.0303	1.6767
16	27.8611	26.5247	1.8099	1.6950
17	27.6919	26.1726	1.7433	1.6621
18	27.6759	27.4920	1.7852	1.6186
19	27.6657	26.3982	1.7757	1.7281
20	27.7239	26.2464	1.7406	1.6794
21	28.0945	26.3413	1.7903	1.6604
22	27.7729	26.9435	1.7340	1.6707
23	28.1143	26.3952	1.7554	1.7057
24	27.7831	26.4012	1.7701	1.6752
25	27.9559	26.3011	1.8119	1.6931
26	28.2036	26.6869	1.7909	1.6682
27	28.0086	26.9060	1.8111	1.7108
28	27.8556	26.7024	1.7702	1.7650
29	28.3404	26.5252	1.7516	1.6890
30	28.0322	27.7842	1.8079	1.6681
31	28.5134	26.4581	1.7444	1.6787
32	28.1892	27.3562	1.7966	1.6928
33	28.0829	26.7050	1.8114	1.7099
34	28.4430	26.8685	1.8707	1.7006
35	28.1640	26.8710	1.8011	1.6976
36	28.8454	26.5781	1.8500	1.6953
37	28.4389	26.7388	1.7986	1.7949
38	28.5883	27.2409	1.8366	1.7011
39	28.4693	27.1756	1.7879	1.7189
40	28.9331	27.2018	1.8111	1.7261
41	28.6134	27.2815	1.7968	1.7064
42	28.7190	27.5326	1.7848	1.7005
43	28.6526	27.3461	1.8746	1.7185
44	29.1826	27.6595	1.8352	1.7713
45	29.2418	27.2559	1.7966	1.7160
TOTAL	**1282.6170**	**1198.768**	**81.4201**	**75.8598**
AVERAGE	**28.5026**	**26.6393**	**1.8093**	**1.6858**

In the study, the average testing time for PCA without CRT is 1.8146 seconds while the average testing time for PCA with CRT is 1.6863 seconds as shown in TABLE 2 and Fig. 7. The experiment concluded that CRT reduces the computational time (for both Training Time and Testing Time) of Principal Component Analysis.

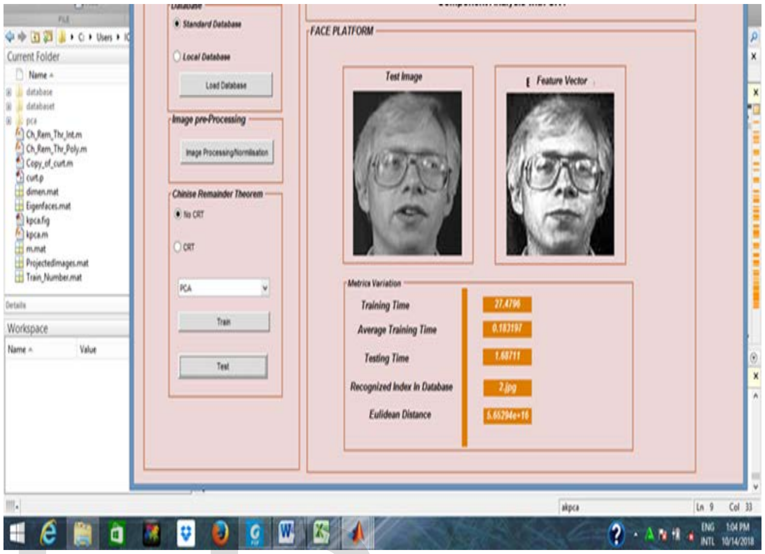

Fig. 4: PCA without CRT on Image 1: Training Time , Testing Time and Recognition

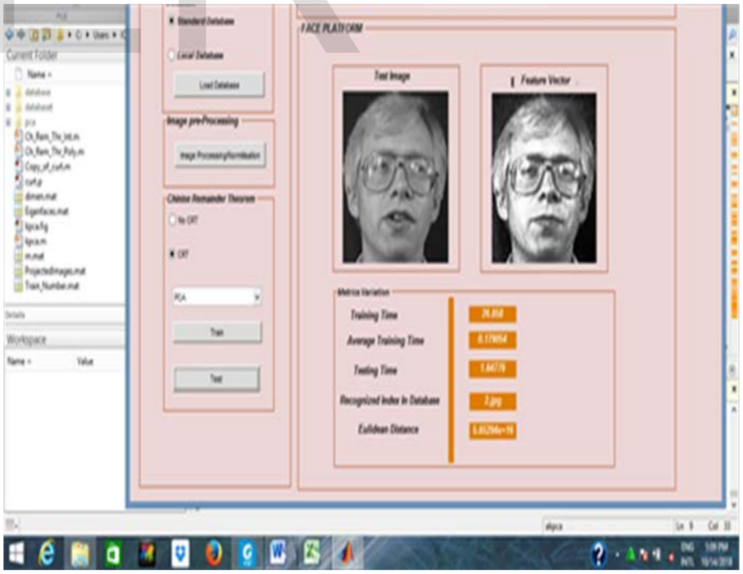

Fig. 5: PCA with CRT on Image 1: Training Time and Testing Time and Recognition Index

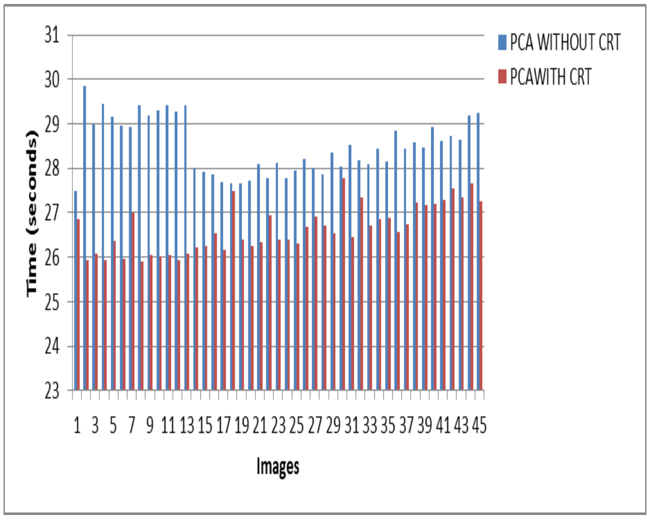

Fig. 6: Training Time for both PCA with and without CRT

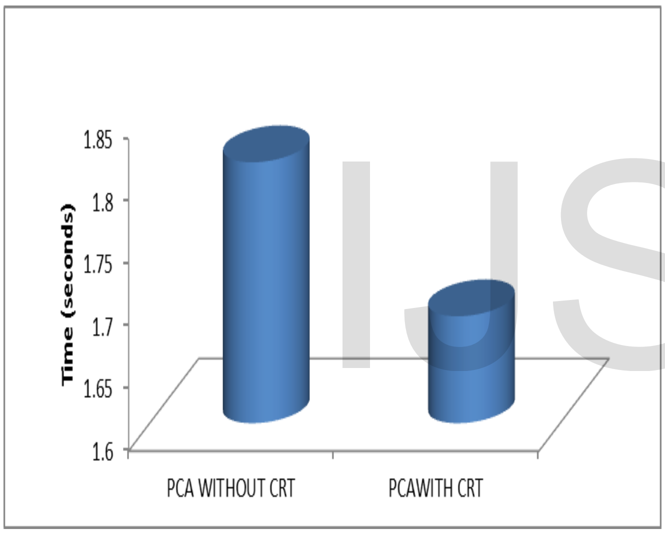

Fig. 7: Average Testing Time for both PCA with and without CRT

TABLE 3 and Fig. 8 & 9 revealed that CRT does not change the recognition accuracy of PCA. It was observed in TABLE 3 that out of the 45 images, 32 images were matched and 13 images did not match when employing CRT with PCA for face recognition and the same result was gotten when CRT was not employ to PCA. TABLE 3 revealed that both PCA with CRT on Image 1 and PCA without CRT on Image 1 have recognized index 2.jpg in database and the image is recognized likewise Fig. 8 shows PCA with CRT on Image 18 and Fig. 9 shows PCA without CRT on Image 18 both having recognized index 30.jpg in database and the image is not recognize.

TABLE 3: Analysis of Image Recognition for both PCA with and without CRT

IMAGE	PCA RECOGNIZED INDEX IN DATABASE WITHOUT CRT	PCA RECOGNIZED INDEX IN DATABASE WITH CRT	IMAGE MATCHED WITHOUT CRT	IMAGE MATCHED WITH CRT
1	2.jpg	2.jpg	YES	YES
2	5.jpg	5.jpg	YES	YES
3	8.jpg	8.jpg	YES	YES
4	10.jpg	10.jpg	YES	YES
5	64.jpg	64.jpg	NO	NO
6	17.jpg	17.jpg	YES	YES
7	19.jpg	19.jpg	YES	YES
8	23.jpg	23.jpg	YES	YES
9	25.jpg	25.jpg	YES	YES
10	28.jpg	28.jpg	YES	YES
11	31.jpg	31.jpg	YES	YES
12	51.jpg	51.jpg	YES	YES
13	37.jpg	37.jpg	YES	YES
14	42.jpg	42.jpg	YES	YES
15	43.jpg	43.jpg	YES	YES
16	48.jpg	48.jpg	YES	YES
17	50.jpg	50.jpg	YES	YES
18	30.jpg	30.jpg	YES	YES
19	55.jpg	55.jpg	YES	YES
20	48.jpg	48.jpg	NO	NO
21	28.jpg	28.jpg	YES	YES
22	41.jpg	41.jpg	YES	YES
23	44.jpg	44.jpg	YES	YES
24	6.jpg	6.jpg	NO	NO
25	75.jpg	75.jpg	YES	YES
26	78.jpg	78.jpg	YES	YES
27	30.jpg	30.jpg	NO	NO
28	17.jpg	17.jpg	NO	NO
29	41.jpg	41.jpg	YES	YES
30	30.jpg	30.jpg	YES	YES
31	30.jpg	30.jpg	YES	YES
32	41.jpg	41.jpg	YES	YES
33	30.jpg	30.jpg	NO	NO
34	8.jpg	8.jpg	YES	YES
35	16.jpg	16.jpg	YES	YES
36	78.jpg	78.jpg	YES	YES
37	48.jpg	48.jpg	NO	NO
38	28.jpg	28.jpg	YES	YES
39	10.jpg	10.jpg	YES	YES
40	45.jpg	45.jpg	NO	NO
41	10.jpg	10.jpg	NO	NO
42	74.jpg	74.jpg	NO	NO
43	30.jpg	30.jpg	NO	NO
44	42.jpg	42.jpg	NO	NO
45	35.jpg	35.jpg	NO	NO

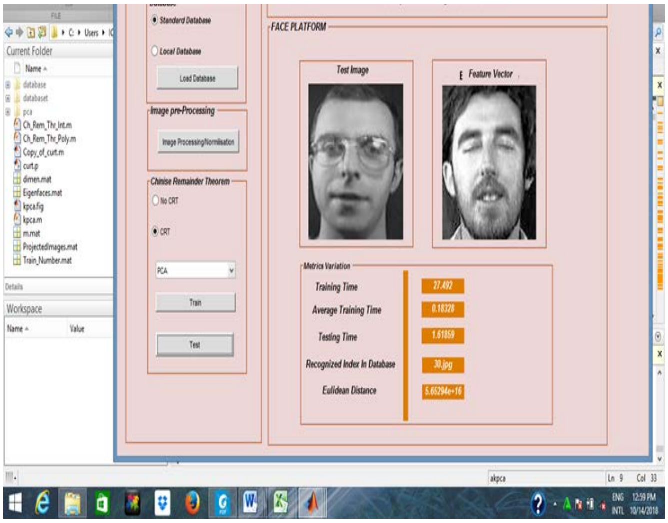

Fig. 8: PCA with CRT on Image 18: Training Time , Testing Time and Recognition Index

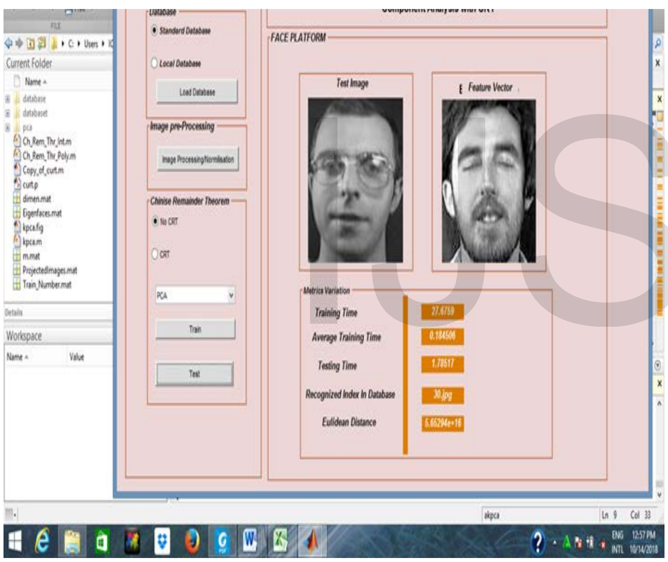

Fig. 9: PCA without CRT on Image 18: Training Time, Testing Time and Recognition Index

5. CONCLUSION

In this study, sequence of input images from YALE face database were trained and tested to determine the effect of CRT on PCA for face recognition. Training time, Testing and recognition index were used as performance metrics. The result obtained from the experiment revealed that PCA uses more Training time and Testing time when CRT was not employ than when employed. Average testing time for PCA without CRT is 1.8146 seconds while the average testing time for PCA with CRT is 1.6863 seconds. It was also deduced that employment of CRT to PCA does not reduce nor increase its recognition accuracy. The same number of images recognized when CRT was not employ with PCA was also recognized when employ with CRT. The future research work will be on employing Mixed Radix Conversion method of RNS on PCA to reduce the computational time of it.

REFERENCES

[1] R.M. Bolle, J.H. Connell, S. Pankanti, N.K. Ratha and K. Senior. *A Guide to Biometrics*, (New York, Springer-Verlag, 2004).

[2] A.G. Youness, A.M. Hamid and T. Mohammad, Adaptive Modified PCA for Face Recognition, *MVA2007 IAPR Conference on Machine Vision Applications*, 3(25), 2007. 126-129.

[3] M.M. Aleix and C.K. Avinash, PCA versus LDA: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2). 2001, 228-233.

[4] A.M. Martinez and A.C. Kak, PCA versus LDA. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(2), 2001, 228-233.

[5] M. Kirby and L. Sirovich, Application of the Karhunen-Loeve procedure for the characterization of human faces. *IEEE Transaction on Pattern and Machine Intelligence*, 12(1), 1990, 103-108.

[6] O.T. Adedeji, E.O. Omidiora, S.O. Olabiyisi and A.A. Adigun, Performance Evaluation of Optimised PCA and Projection Combined PCA methods in Facial Images. *Journal of Computations & Modelling*, 2(3), 2012, 17-29.

[7] J.O. Aluko, E.O. Omidiora, A.B. Adetunji and O.A. Odeniyi, Performance Evaluation Of Selected Principal Component Analysis-Based Techniques for Face Image Recognition, *International Journal Of Scientific & Technology Research*, 4(01), 2015, 35-41.

[8] K. Jain and S. Singh, Performance Evaluation of Face Recognition Using PCA. *International Journal of Information Technology and Knowledge Management*, 4(2), 2011, 427-430.

[9] K.A. Gbolagade, *Effective Reverse Conversion in Residue Number System Processors*. PhD Thesis. The Netherlands, 2010.

[10] A. Omondi and B. Premkumar, Residue Number Systems: Theory and Implementation. *Advances in Computer Science and Engineering*, 2. Imperial College Press. London, 2007, 2.

[11] N. Szabo and R. Tanaka, *Residue arithmetic and its application to computer technology* (New York, MC-Graw-Hill, 1967)

[12] S. Sahaana, and B. Peter, DROP: A Workload-Aware Optimizer for Dimensionality Reduction. In *International Workshop on Data Management for End-to-End Machine Learning (DEEM'30), Amsterdam, Netherlands.* ACM, New York, NY, USA, 2019.

[13] V. Gassenbauer, J. Křivánek, K. Bouatouch, C. Bouville and M. Ribardière, Improving Performance and Accuracy of Local PCA. *Computer Graphics Forum*. The Eurographics Association and Blackwell Publishing Ltd., USA, 30(7), 2011.

[14] The Yale database, Available: http://cvc.yale.edu