International Journal of Discrete Mathematics
2019; 4(1): 1-7
http://www.sciencepublishinggroup.com/j/dmath
doi: 10.11648/j.dmath.20190401.11
ISSN: 2578-9244 (Print); ISSN: 2578-9252 (Online)

SciencePG
Science Publishing Group

Methodology Article

Reducing Computational Time of Principal Component Analysis with Chinese Remainder Theorem

Madandola Tajudeen Niyi[1], Gbolagade Kazeem Alagbe[2]

[1]Department of Computer Science, Kwara State College of Education, Oro, Nigeria
[2]Department of Computer Science, Kwara State University, Malete, Nigeria

Email address:
nmadandola@yahoo.com (M. T. Niyi). kazeem.gbolagade@kwasu.edu.ng (G. K. Alagbe)

To cite this article:
Madandola Tajudeen Niyi, Gbolagade Kazeem Alagbe. Reducing Computational Time of Principal Component Analysis with Chinese Remainder Theorem. *International Journal of Discrete Mathematics.* Vol. 4, No. 1, 2019, pp. 1-7. doi: 10.11648/j.dmath.20190401.11

Received: February 1, 2019; **Accepted**: March 12, 2019; **Published**: March 30, 2019

Abstract: It is of paramount importance to establish an identity of citizenry to curb criminalities. Principal Component Analysis (PCA) which is one of the foremost methods for feature extraction and feature selection is adopted for identification and authentication of people. The computational time used by PCA is too much and Chinese Remainder Theorem was employed to reduce its computational time. TOAM database was setup which contained 120 facial images of 40 persons frontal faces with 3 images of each individual. 80 images were used for training while 40 were used for testing. Training time and testing time were used as performance metrics to determine the effect of CRT on PCA in terms of computational time. The experimenal results indicated an average training time of 13.5128 seconds and average testing time of 1.5475 second for PCA while PCA-CRT average training time is 13.2387 seconds and average testing time of 1.5185 seconds. Column chart was used to show the graphical relationship between PCA and PCA-CRT Training time and testing time. The research revealed that CRT reduce PCA computational time.

Keywords: Dimensionality Reduction, Chinese Remainder Theorem, Eigenface, Training Time, Database

1. Introduction

Reference [8] pointed out that establishing identity of an individual is of paramount importance in extremely interrelated organization. It is of greatest important because of increase in daily criminal acts in the world today especially in Nigeria. The time required to identify and authenticate such individual needs to be minimized for effectiveness of the system. Therefore, dimensionality algorithms computational time should be reduced. Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) are two foremost methods used for feature extraction and feature selection in the appearance-based methods. It has been discussed by Reference [1] that when the training dataset is small, PCA can outperform LDA, and also that PCA is less sensitive to different training datasets. Reference [6] explained that PCA and LDA are the two most popular techniques usually used for dimensionality reduction. Reference [5] were among the first to use PCA with face images and found that it effectively and efficiently represents pictures of faces into its eigenface components. Principal Component Analysis is a powerful dimensionality reduction algorithm and it is reliable in face recognition system but its computational time needs to be reduced. Reference [4] described PCA as a technique for reducing the dimensionality of datasets, increasing interpretability but at the same time minimizing information loss.

Reference [9] adopted the use Compute Unified Device Architecture (CUDA) to tackle the problem of enormous data processing time that resulted as a consequence of very large data extracted from the face images. In order to reduce the execution time of these operations, parallel programming techniques are used. Reference [2] focused on Improving Performance and Accuracy of Local Principal Component Analysis (LPCA). LPCA also referred to as Clustered PCA (CPCA). A novel SortCluster LPCA (SC-LPCA) was recommended. The arrangement of SC-LPCA algorithm is related to the SortMeans algorithm [7]. The new algorithm (SortCluster LPCA) was compared with the original LPCA for compression of Pattern

Recognition Technique (PRT) and Bidirectional texture function (BTF) datasets. Both algorithms were implemented in C++ using Intel MKL library for matrix computations. All measurements were performed on a PC with Intel Xeon W3540, 2.93 GHz and 14GB RAM. The new algorithm considerably reduces the cost of the point-cluster classification stage, achieving a speed-up of up to 20s. However, instead of computing distances with respect to the clusters' centroids, distances were computed with respect to affine subspaces.

Having since other researchers' efforts in reducing the computational time of PCA. Chinese Remainder Theorem (CRT) was introduced to PCA as shown in Figure 3. CRT which is one of the forms of Residue Number System is appropriate because the computation can be parallelized. In RNS, addition, subtraction and multiplication are inherently carry-free. As a result of this carry-free property, addition, subtraction and multiplication can be executed in CRT in less time than would be required for performing parallel binary addition in a computer of equal component operating speed and number range. A face database (TOAM database) will be set up for the purpose of the research work. Euclidean distance will be used as a classifier for incoming test data.

2. Principal Component Analysis

Principal Component Analysis is the basic linear technique for dimensionality reduction which implements a linear mapping (as shown in Figure 1) of the data to a lower-dimensional space in such a method that the discrepancy of the data in the low-dimensional representation is maximized. Traditionally, the covariance matrix of the data is constructed and the eigenvectors on this matrix are computed. PCA tries to find a low-dimensional linear subspace that the data are confined to. Principal Component Analysis (PCA) algorithm involves substantial mathematical operations with matrices. The PCA involves two (2) stages: training and recognition. In the training dataset, there is an eigenvector for each face as shown in Table 1. Also, in the recognition stage the eigenvectors with highest values are chosen and treated as shown in Table 2. A threshold value is used for comparison in the recognition stage.

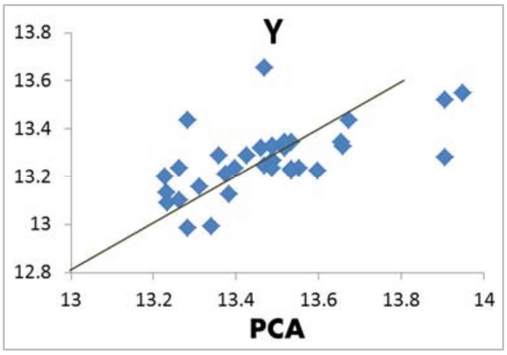

Figure 1. *PCA implementing a linear mapping of the data [11].*

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 {D1, D2,…, DM} be the training data set. The average Avg is defined by:

$$\text{Avg} = \frac{1}{M}\sum_{i=1}^{M} Di$$

2. Distinctive element in the training data set diverges from by the vector *Yi=Di-Avg*. The covariance matrix obtained as:

$$\text{Cov} = \frac{1}{M}\sum_{i=1}^{M} Yi.Yi^{T}$$

3. Pick M' significant eigenvectors of and compute the weight vectors *Wik* the training data set, where *k* varies from 1 to M'.

$$W_{ik} = E^{T}_{k}.(D_i - \text{Avg}), \forall\, i, k$$

Table 1. *PCA Algorithm Calculation Steps of Feature Extraction [9].*

Algorithm	Description	Explanation
Normalization	$X' = \frac{x-min_A}{max_{A}-min_A}$	X represents the color value of a pixel in the image, min_A represents the smallest color value in the face (i.e. image), max_A represents the biggest color value in the face (i.e. image).
Mean Calculation	$\Psi = \frac{1}{M}\sum_{n=1}^{M} \Gamma n$	$\Gamma_1, \Gamma_2, .. \Gamma_M$ column matris corresponding to each face in the training set, if the size of the face image is N x N then each column matris Γ will have the size N^2 x 1, M number of faces in the training set. Ψ mean value of the set.
Φ calculation	$\Phi = \Gamma_i - \Psi$	Φ_i the difference of the face from the mean value.
A Transpose Calculation	A^T	$A = [\Phi_1, \Phi_2, \dots, \Phi_M], \Phi_i$ fort he size N^2x1, The A matris size becomes N^2xM.
L Calculation	$L = A^T A$	L reduced matris (sizewise) calculated instead of covariance matris, multiplication of MxN^2 with N^2xM sized matris produces MxM sized matris.
Eigenvector Calculation	$L . v = \lambda v$	v is eigenvector and λ is eigenvalue
Eigenface Calculation	$u_i = \sum_{k=1}^{M} v_{ik\Phi_k}$	U is eigenface, i = 1,…,M
Feature Vector Calculation	$w_k = u^T_k(\Gamma - \Psi)$	W is the feature vector

Table 2. *PCA Algorithm Calculation Steps of Face Recognition [9].*

Algorithm	Description	Explanation
Normalization	$x' = \frac{x-min_A}{max_{A}-min_A}$	The color value of the face to be recognized normalized to between 0 - 1
Difference from mean	$\Phi_T = \Gamma_T - \Psi =$	Its difference from the mean face is calculated
Feature vector extraction	$w_{t=U.\Phi_T}$	The feature vector is calculated using the eigenvectors
Distance Control	$d_i = \sum_{j=i}^{M} \lvert w_T(j) - w_i(j) \rvert$	The distance between the feature vectors are calculated for a successful matching, i = 1,2…,N

3. Chinese Reminder Theorem

The Chinese Remainder Theorem (CRT) in addition to the theory of RNS, was set forth in the 19th century by Carl Friedrich Gauss in his celebrated Disquisitions Arithmetical [10], [3]. CRT is a proposition of number theory that expresses that if one knows the remainders of the Euclidean division of an integer *n* by several integers, then one can establish distinctively the remainder of the division of *n* by the product of these integers, under the condition that the divisors are pairwise coprime. In the third century A. D. a Chinese mathematician named Sun Zhu first described CRT. The CRT makes it feasible to reduce modular calculations with large moduli to similar calculations for each of the factors of the modulus.

Reference [12] pined the traditional CRT as follows:
for a moduli set $\{m_1, m_2, m_3, \ldots, m_k\}$ with the dynamic range M = $\prod_{i=1}^{k} m_i$, the residue number ($x_1, x_2, x_3, \ldots, x_k$) can be changed into the decimal number X, as follows:

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

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.

4. Research Database

A database tagged “TOAM database” was setup which contains 120 facial images of 40 persons frontal faces with 3 images of each individual under different lighting, facial expressions, occulations, environment and time. 80 images will be used for training while 40 will be used for testing as shown in Table 3 and Figure 2.

Table 3. *Analysis of images used in the TOAM database.*

Number of persons	40
Number of sample per persons	03
Number of Total sample	120
Number of Training set	80
Number of Testing sample	40

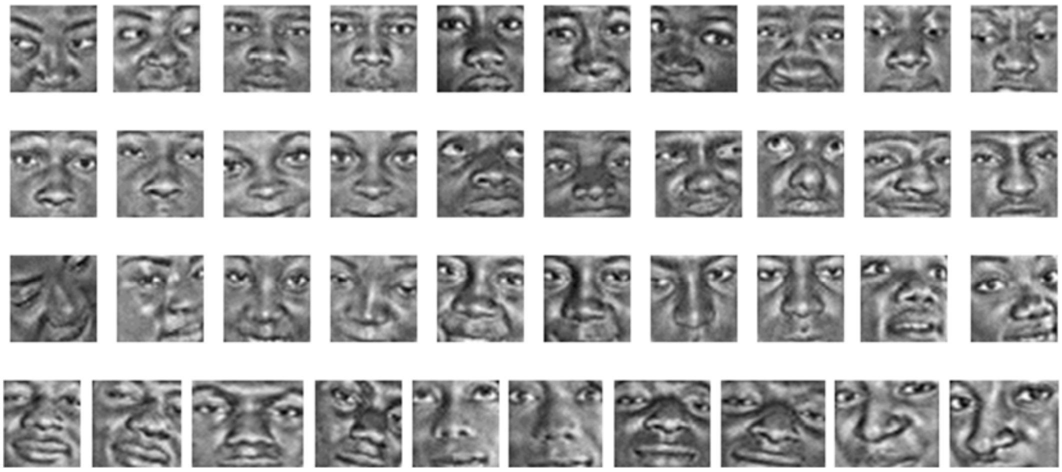

Figure 2. *Some of TOAM database images used for training the database.*

5. Research Methodology

i. Develop a Research framework as shown in figure 3
ii. Develop algorithm that combine CRT with PCA (PCA-CRT)
iii. Comparison of performance metrics (Training time, and testing time) on PCA and PCA-CRT
iv. Using column chart to compare results

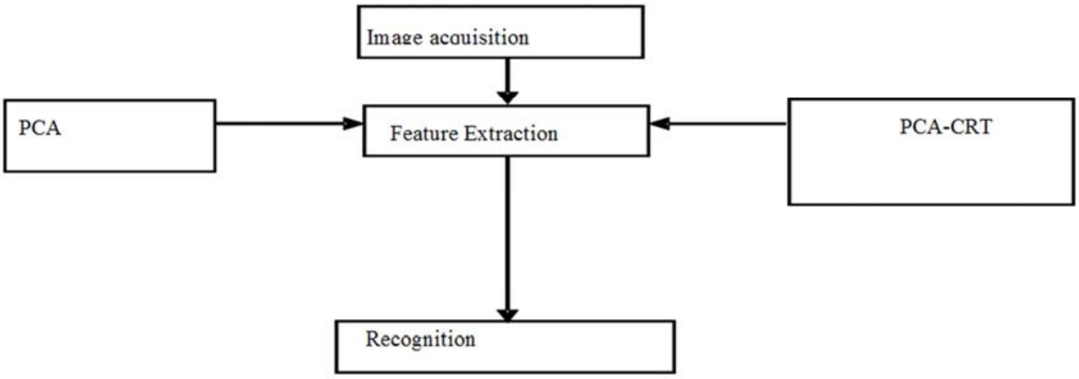

Figure 3. *Research Framework.*

The proposed system consists of number of modules: image acquisition, feature extraction, and recognition. In this research MATLAB R2015a was used to implement effect of PCA and PCA-CRT as shown in Figure 4 on Intel (R) Celeron (R) CPU with 1.60GHz Processor speed.

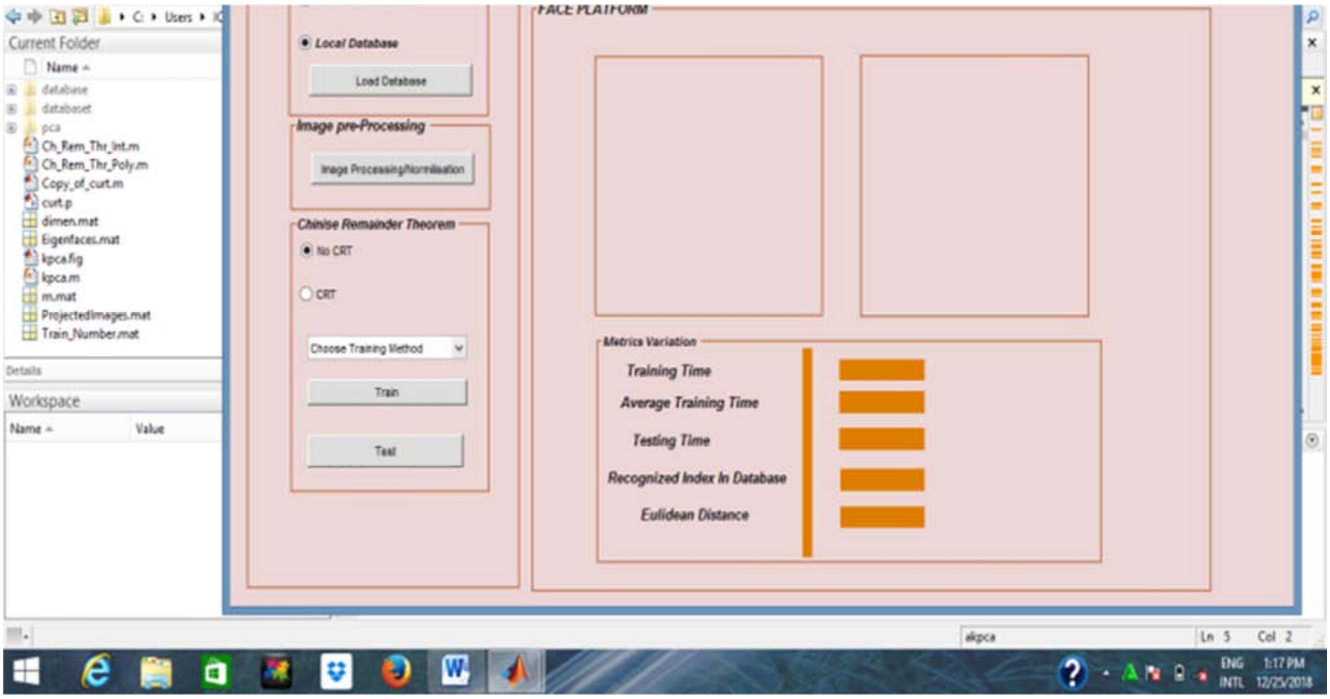

Figure 4. Proposed system for Implementing PCA and PCA-CRT.

6. Results and Discussion

6.1. Comparison of PCA and PCA-CRT Testing Time

Table 4. Testing Time Performance of both PCA and PCA-CRT.

IMAGE	PCA TESTING TIME (Seconds)	PCA-CRT TESTING TIME (Seconds)	IMAGE	PCA TESTING TIME (Seconds)	PCA-CRT TESTING TIME (Seconds)
1	1.6323	1.5486	22	1.5374	1.5156
2	1.5212	1.5160	23	1.5468	1.5304
3	1.5536	1.5302	24	1.5342	1.5200
4	1.5065	1.4899	25	1.5396	1.5134
5	1.5333	1.5200	26	1.5572	1.5045
6	1.5881	1.5398	27	1.5809	1.5054
7	1.5204	1.5197	28	1.5747	1.5395
8	1.5088	1.4989	29	1.5240	1.4982
9	1.5432	1.5222	30	1.5565	1.5322
10	1.5360	1.5276	31	1.5929	1.5557
11	1.5242	1.4679	32	1.5386	1.4900
12	1.5538	1.5513	33	1.5257	1.5134
13	1.4931	1.4888	34	1.5680	1.5322
14	1.5349	1.5111	35	1.5701	1.5444
15	1.5439	1.5166	36	1.5853	1.5509
16	1.5497	1.5109	37	1.5665	1.5134
17	1.5257	1.5121	38	1.5676	1.5178
18	1.5435	1.4789	39	1.5568	1.5335
19	1.5237	1.5225	40	1.5560	1.5268
20	1.5453	1.4966	TOTAL	61.9014	60.7406
21	1.5414	1.5337	AVERAGE	1.5475	1.5185

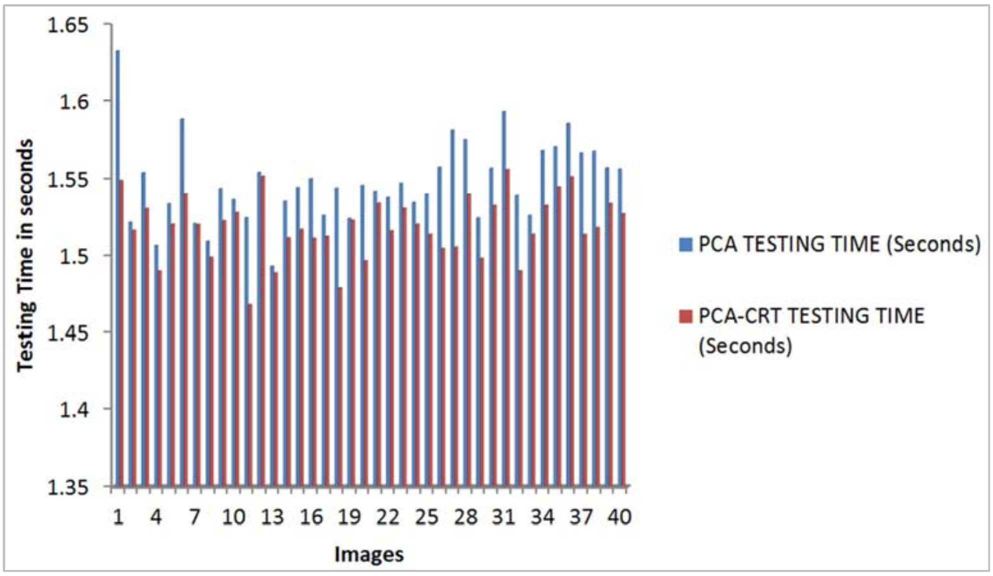

Figure 5. *Testing Time of PCA vs PCA-CRT.*

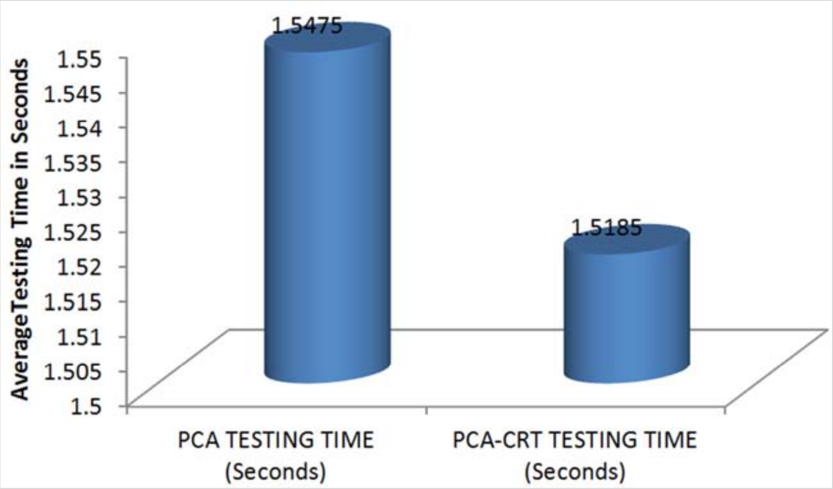

Figure 6. *Average Testing Time of PCA vs PCA-CRT.*

Critical observation of Table 4 and Figure 5 revealed that each of the images used spent more PCA Testing Time than PCA-CRT. Using image 20 as case study, PCA uses 1.5453 seconds Testing Time while PCA-CRT uses 1.4966 seconds. The Total Testing time for PCA is 61.9014s and PCA-CRT Total Testing Time is 60.7406s as shown in Table 3. Figure 6 shown pictorially Average Testing Time for PCA of 1.5475 seconds and PCA-CRT of 1.5185 seconds. The experimental results indicated reduction in Testing time of PCA as a result of employing CRT.

6.2. Comparison of PCA and PCA-CRT Training Time

Table 5. *Training Time Performance of both PCA and PCA-CRT.*

IMAGE	PCA TRAINING TIME (Seconds)	PCA-CRT TRAINING TIME (Seconds)	IMAGE	PCA TRAINING TIME (Seconds)	PCA-CRT TRAINING TIME (Seconds)
1	13.4357	12.7146	22	13.4907	13.2654
2	13.4860	12.8783	23	13.5329	13.2315
3	13.3402	12.9886	24	13.5176	13.3211
4	13.2610	13.1009	25	13.4696	13.2444
5	13.2622	13.2352	26	13.9067	13.5229
6	13.3843	13.1243	27	13.6725	13.4342
7	13.2277	13.2001	28	13.5546	13.2345
8	13.4613	13.3211	29	13.6596	13.3246
9	13.2307	13.1320	30	13.5981	13.2228

IMAGE	PCA TRAINING TIME (Seconds)	PCA-CRT TRAINING TIME (Seconds)	IMAGE	PCA TRAINING TIME (Seconds)	PCA-CRT TRAINING TIME (Seconds)
10	13.4249	13.2882	31	13.6565	13.3434
11	13.4873	13.2657	32	13.5530	13.2388
12	13.2343	13.0900	33	13.6573	13.4668
13	13.5341	13.3458	34	13.8141	13.4456
14	13.4626	13.3222	35	13.6960	13.3267
15	13.3118	13.1567	36	13.6682	13.1299
16	13.9499	13.5493	37	13.6569	13.3368
17	13.3768	13.2111	38	13.5756	13.1296
18	13.3579	13.2889	39	13.7193	13.3006
19	13.3976	13.2355	40	13.7175	13.2666
20	13.2843	12.9872	TOTAL	540.5134	529.5474
21	13.4861	13.3255	AVERAGE	13.5128	13.2387

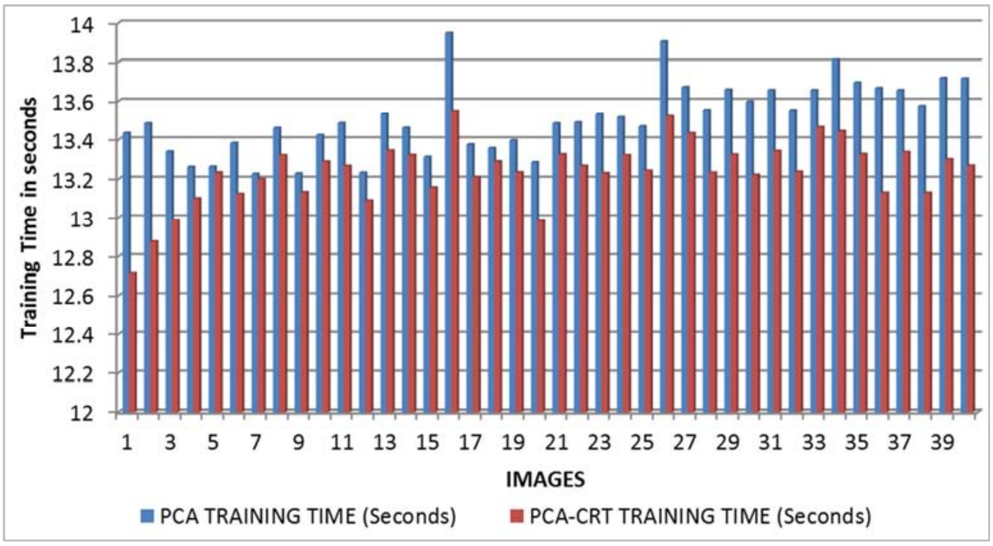

Figure 7. *Training Time of PCA vs PCA-CRT.*

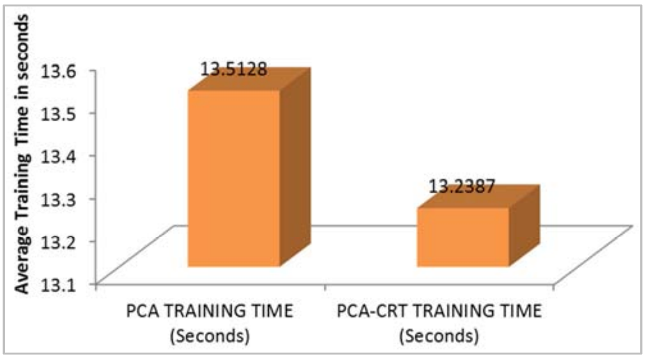

Figure 8. *Average Training Time of PCA vs PCA-CRT.*

Table 5 and Figure 7 shown variation of Training Time used by both PCA and PCA-CRT on each image. It was deduced that each of the image Training Time used by PCA-CRT is lesser than those of PCA. Training time for image 1 for PCA is 13.4357s while image 1 Training time for PCA-CRT is 12.7146s. The Table 5 also had shown Total Training Time of 540.5134 seconds for PCA and 529.5474 seconds for PCA-CRT indicating a more Computational time for PCA than PCA-CRT. Figure 8 shown distinctions between the average Training time used by PCA and PCA-CRT. The average Training time used by PCA is 13.5128 seconds while average Training time used by PCA-CRT is 13.2387 seconds.

7. Conclusion

In this paper, chain of input images from setup database tagged "TOAM database" were trained and tested to determine the effect of CRT on computational time of PCA for face recognition. Training and Testing time were used as performance metrics. The investigation shown that PCA uses more Training Time and Testing time when CRT was not employed than when employed. PCA-CRT average testing time is 1.5185 seconds while the average testing time for PCA is 1.5475 seconds. The experimental result reveals a total Training time of 540.5134 seconds for PCA while total Training time used by PCA-CRT is 529.5474 seconds.

References

[1] Aleix M. M. and Avinash C. K. (2001). IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 23, No.2, pp. 228-233, 2001 PCA versus LD A.

[2] Gassenbauer , V., Krivánek, J., Bouatouch, K., Bouville, C. and Ribardière, M. (2011). Improving Performance and Accuracy of Local PCA. Computer Graphics Forum. The Eurographics Association and Blackwell Publishing Ltd, 9600 Garsington Road, Oxford OX4 2DQ, UK and 350 Main Street, Malden, MA 02148, USA.

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

[4] Jolliffe I. T and Cadima J. (2016). Principal component analysis: a review and recent developments. Phil. Trans. R. Soc. A 374: 20150202. http://dx.doi.org/10.1098/rsta.2015.0202.

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

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

[7] Phillips, S. J. (2002). Acceleration of K-means and related clustering algorithms. In *Algorithm Engineering and Experiments (ALENEX)*, pp. 166–177.

[8] Ross A. (2007). “An Introduction to Multibiometrics”. 15th European Signal Processing Conference (EUSIPCO), Poznan, Poland.

[9] Salih, G., Kadriye Ö. and Abdullah, Ç. (2016). CUDA Based Speed Optimization of the PCA Algorithm. *TEM Journal* Volume 5 / Number 2 / 152–159.

[10] Soderstrand, M. A., Jenkins, W. K., Jullien, G. A. and Taylor, F. J.(1986). Residue Number System Arithmetic: Modern Applications in Digital Signal Processing. IEEE press, Piscataway, NJ, USA.

[11] Symeonidis, et al. (2010): What are the advantages of kernel PCA over standard PCA retrieved from https://stats.stackexchange.com/questions/94463/what-are-the-advantages-of-kernel-PCA-over-standard-PCA

[12] Szabo, N. and Tanaka, R. (1967). Residue Arithmetic and its Application to Computer Technology. MC-Graw-Hill, New York.