Bachelor of Science in Computer Science and Engineering

Compression of Large-Scale Image Dataset by Dimensionality Reduction using PCA & Color Quantization Using K-means Clustering

By

Rushrukh Rayan

ID: 1304040

October, 2018

Department of Computer Science & Engineering

Chittagong University of Engineering & Technology

Chittagong-4349, Bangladesh.

Compression of Large-Scale Image Dataset by Dimensionality Reduction using PCA & Color Quantization Using K-means Clustering

This thesis is submitted in partial fulfillment of the requirement for the degree of Bachelor of Science in Computer Science & Engineering

By

Rushrukh Rayan

ID: 1304040

Supervised by

Dr. Asaduzzaman

Professor

Department of Computer Science & Engineering (CSE)

Chittagong University of Engineering & Technology (CUET)

Department of Computer Science & Engineering

Chittagong University of Engineering & Technology

Chittagong-4349, Bangladesh.

The thesis titled **"Compression of Large-Scale Image Dataset by Dimensionality Reduction using PCA & Color Quantization Using K-means Clustering"** submitted by Roll No. 1304040, Session 2016-2017 has been accepted as satisfactory in fulfillment of the requirement for the degree of Bachelor of Science in Computer Science & Engineering (CSE) as B.Sc. Engineering to be awarded by the Chittagong University of Engineering & Technology (CUET).

Board of Examiners

1. _____ Chairman

Dr. Asaduzzaman
Professor
Department of Computer Science & Engineering (CSE)
Chittagong University of Engineering & Technology (CUET)

2. _____ Member

Dr. Mohammad Shamsul Arefin (Ex-officio)
Professor & Head
Department of Computer Science & Engineering (CSE)
Chittagong University of Engineering & Technology (CUET)

3. _____ Member

(External)
Abu Hasnat Mohammad Ashfak Habib
Associate Professor
Department of Computer Science & Engineering (CSE)
Chittagong University of Engineering & Technology (CUET)

Statement of Originality

It is hereby declared that the contents of this project are original and any part of it has not been submitted elsewhere for the award of any degree or diploma.

Signature of the Candidate

Date:

Acknowledgment

First of all, I am grateful to almighty Allah for enabling me to complete this thesis successfully. Thereafter, my sincerest thanks and gratitude to my honorable project supervisor Dr. Asaduzzaman, Professor of Computer Science & Engineering, Chittagong University of Engineering & Technology, for his valuable suggestions, positive advises, encouragement and sincere guidance throughout my thesis work. I also convey my special thanks and gratitude to all of respected faculty members of the department. I would like to thank all of my acquaintances & the staffs of the department for their valuable suggestions and assistance. Finally, I would like to thank my parents for their steady support during my study period.

Abstract

In this modern era of digitalization along with unprecedented internet facilities accompanied by smart devices, images are captured and shared in a bulk extent. These large amount of images have to be stored in the database for storage purpose & further processing. Often these images contain feature redundancy which can be eliminated to an ideal extent that helps reducing the image file size. Color quantization is another criteria to restrict the distinct amount of colors represented in the image. Efficient storage & further processing of these images yield compression of the data. In this dissertation work, an approach is proposed to compress a large-scale image dataset. For dimensionality reduction, Principal Component Analysis is implemented to project the data onto a lower dimensional space & color quantization is achieved through implementing K-means clustering algorithm. By combining these two methods, a new method is implemented to compress the dataset. The core challenges emerged in this dissertation work is to determine the right number of dimensions onto which to project the data & to determine the right number of clusters to perform color quantization. Previously, a system to compress a large-scale image dataset was obtained by implementing the two algorithms alone. In this thesis, a new method by combing the two algorithms has been proposed. Compression ratio for the same dataset, obtained from K-means clustering algorithm is 1.546, from PCA is 1.604. By implementing the proposed method, we obtain a compression ratio of 1.648.

Contents

List of Figures

Chapter 1

Introduction

Image compression is typically a process that results in reduction of the image file size, utilizing the visual grasp in such a way that provides resourceful insights to the image components in sort of a numerical data. In order to deal with large-scale datasets, where the number of features can be very gigantic, machine learning can be a very constructive approach that can provide with a model, by dealing with the inputs through supervised or unsupervised learning algorithms & produce a fully data-driven decision. Moreover, with unsupervised learning, speculations can be inferred from dataset consisting of unlabeled data usually by searching for hidden patterns or clusters in the dataset, which actually remains hidden to the human eyes before analyzing the data.

Quantized large-scale image datasets has ubiquitous applications such as following:

For instance, for a facial recognition system of a university, the dataset to match with the test image can be pretty large, containing up to 4000-5000 images. These images are stored in a database & in the process, is compared to the test images obtained from the camera placed at the entrance. These images often can be large in file size than required, can contain data redundancy in them. To obtain a less complex & efficient facial recognition system that takes less amount of a storage, a large-scale image dataset compression method can be a great data preprocessing step. Not only facial recognition, in applications such as action recognition, various object detection & recognition, handwriting & character recognition, where training a model with large-scale image dataset is essential, the compression method can be of great help.

In this modern era where large-scale data transaction and consumption have become a consistent event, the need of data compression is hence, indispensable. With the advent of modern technologies & smart devices such as phones, tablets, digital cameras along with ever growing facilities of internet, the amount of image produced, processed & transmitted is enormous which is to be reduced in terms of size; in order to increase storage efficiency, to reduce time delays in case of data transmission & to achieve an upper hand in many of the processing methods of these images.

1.1 Compression of Large-Scale Image Dataset

A brief overview of compression of a large-scale image dataset has been discussed in this section, including the background and present state of the problem along with the challenges, motivation, objectives, scope & limitations of the dissertation work.

Image compression can be defined as the process of encoding image using a representation that reduces the size of data to an acceptable extent which is only possible if there is redundancy in the original data. A lossy compression, which is generally devised by compressing a given set of values to a lesser set of values, so as to minimize distinct number of colors for instance, is called Quantization.

A quantized, redundancy-divested dataset, especially when it is regarded in the large-scale can flatten a lot of complexities that could have emerged if not quantized. For instance, face recognition has to deal with precise pixel values of a large dataset. It becomes prerequisite to quantize this bulk amount of images in order to increase efficiency along with diminishing complexities. Hence, it yields no argument that these large-scale datasets need to be compressed to an ideal extent before undergoing further processing.

For unsupervised learning in the area of machine learning, the datasets not necessarily need to be labeled, high-quality datasets can require a robust & costly procedure [1]. Thus, it motivates us to design a system that compresses large-scale image dataset. To compress a large-scale dataset, a generally emerged framework works as below-

- Preprocessing of dataset which may include cropping images into a specific ratio.

- Dimensionality Reduction

- Color Quantization

- Result Comparison & Analysis

1.2 Background & Present State of the Problem

This dissertation work shares similarity with below discussed systems in works related to compression of images:

K-means clustering algorithm produced a quantized color image represented by the contours while in processing or clustering part [2], here, Genevieve & Erees used a captured image as the input data to undergo K-Means color quantization algorithm to quantize the image into discrete number of colors. What K-means essentially does to the image here is to separate samples in n-color groups of equal variance.

Herve Abdi & Lynne J. Williams in their work [3] described Principal component analysis (PCA) as "A multivariate technique that analyzes a data table in which observations are described by several inter-correlated quantitative dependent variables. Its goal is to extract the important information from the table, to represent it as a set of new orthogonal variables called principal components, and to display the pattern of similarity of the observations and of the variables as points in maps".

Nitika Sharma & Kriti Saroha used dimensionality reduction of Breast Cancer dataset, failing to do so could have resulted in "Curse of Dimensionality" which occurs due to presence of large number of dimensions in a dataset. This problem leads to reduced accuracy of machine learning classifiers because of presence of many insignificant and irrelevant dimensions or features in the dataset [4].

A closely related work with this dissertation would be Chih-Wen & Jyh-Horng's work [5]. They used PCA to transform the data set into a lower dimensional feature space & K-means to partition the training set into clusters. What differs between their work & this dissertation is they have built the system for the purpose of compression of a single image while in this case, the implementation largely focuses on the large-scale dataset. To summarize, this method is proposing a system to compress a large-scale image dataset instead of being implemented to just a single image.

Turgay Celik in his work developed a novel technique for unsupervised change detection in multi-temporal satellite images using Principal Component Analysis (PCA) and K-Means Clustering [6]. Analyzing the difference image of two satellite images acquired from the same area coverage but at two different time instances, the non-overlapping blocks of the difference image are used to

extract eigenvectors by applying PCA, after that, a feature vector for each pixel of the difference image is extracted by projecting its h × h neighborhood data onto eigenvector space & finally the feature vector space is clustered into two clusters using k-means algorithm [7].

In contrast to these research, we intend to design a system in which dimensionality reduction & color quantization of a large-scale image dataset is obtained by implementing Principal Component Analysis & K-means clustering algorithm respectively.

1.3 Challenges

The main challenges in developing such system are to choose the right number of clusters for color quantization & to choose the right number of components for dimensionality reduction. Furthermore, to achieve the best possible results, optimization objectives such as cost functions, projection errors etc. are to be minimized. Also, there may be some data preprocessing included, if required, such as feature scaling & mean normalization.

1.4 Motivation of the Research

The system that helps compress a large-scale image dataset can be employed in ubiquitous domains such as preparing dataset to better suit Facial Recognition, Action Recognition, Object Detection & Recognition, Handwriting & Character Recognition, Aerial Images and several other fields of computer vision & deep learning & so on. Regardless of the further processing, it is important to come up with new compression techniques so as to evolve the data compression method domain.

Our goal in this dissertation work is to achieve the best possible results to make the system as efficient as possible, which is believed to provide a substantial amount of insights that is required to develop a new method to compress a large-scale image dataset in the field of Machine Learning.

1.5 Objective

The key objectives of this dissertation work is to design a system which essentially can compress a large-scale image dataset. The key objective and possible outcomes of this work are mentioned below:

- To reduce dimensionality of large-scale image dataset using Principal Component Analysis.
- To obtain color quantization of large-scale image dataset using K-means.
- To evaluate the system efficiency with the aforementioned objectives individually.

1.6 Scope & Limitations

The proposed method works fine during the whole process except when the number of clusters is being increased, the time required to complete the K-means algorithm on the dataset increases substantially. Furthermore, if the number of dimensions onto which we want to project the data on is relatively small, the overall image quality drops from the standard.

1.7 Organization of Report

The rest of the report is organized as follows: in chapter 2, dimensionality reduction, Principal Component Analysis, color quantization, Mean Normalization, Covariance Matrix, Eigenvector of Covariance Matrix, K-means clustering algorithm are explained along with background & current state of the problem. In chapter 3, the proposed methodology is explained with output of each implementation state. Chapter 4 consists of sample input & output followed by rigorous experimental analysis. Later in chapter 5, future work suggestions have been represented.

Conclusion

In this chapter, we mainly focused on getting acknowledged with the motivations, objectives, challenges, scopes & limitations in order have a primary insight to the problem that is later solved in the dissertation work.

Chapter 2

Literature Review & Definitions

2.1 Introduction

To develop a system for compressing large-scale image dataset by means of dimensionality reduction using Principal Component Analysis algorithm along with color quantization by deploying K-Means clustering algorithm, a systematic literature review is undertaken that is described later in this chapter.

This chapter is dedicated to the basic concepts of Lossy Image Compression, Dimensionality Reduction, Color Quantization, Principal Component Analysis for achieving reduced dimensionality & K-means for achieving color quantization.

2.2 Lossy Image Compression

An unalterable process that involves permanent loss of a particular portion of detail from an image which is being decided based on criteria obtained from performing a compression method on the image. In digital images, often there is found redundancy, in other words, correlated features diminishing which, it is possible to lessen the file size with a fair trade-off with image quality. This is phenomenon is referred to as Lossy Image Compression.

Figure 2.1 (a)

Figure 2.1 (b)

Figure 2. 1 Lossy Image Compression (a) Original Image (10.5KB), (b) Compressed Image (6.46KB)

2.3 Dimensionality Reduction

The sample input images are first read & converted into a data matrix where, if the number of input samples is n and each sample is of size p x q, the data matrix dimension would be n x (p x q) where each row is a sample input and each column represents a feature. Sometimes, the number of features can be very large, where a lot of these features may be correlated, thus, considered as redundant. The higher the number of features, the harder it is to obtain visual perception from the training set & to work on it. We wish to project the data onto a lower dimensional space m, where m is certainly less than (p x q).

Formally, dimensionality reduction would be the process of reducing the number of random variables under consideration regarding expected output, by projecting the data onto a lower dimensional surface which results in compressing the file size along with enhancement of visual perception.

2.4 Color Quantization

Formally, color quantization would be the process that involves reducing the number of distinct colors that are being used in an image, producing an output that practically retains similarity as close as possible to the input image, but reducing the number of colors used in the image.

2.5 Principal Component Analysis

Principal Component Analysis (PCA) is performed with two objectives, first would be dimensionality reduction for the purpose of compression of the data & latter would be to have a visual perception of the data which can further provide information that originally was hidden to the eyes. What it essentially does is to find a lower dimensional surface onto which to project the data so as to minimize the projection error. If originally the dataset incorporates n-dimensions, PCA will reduce data to k-dimensions, with k<n being classified.

Initially, in this algorithm, Covariance Matrix is computed. Then, Eigenvectors of the Covariance Matrix is computed using Singular Value Decomposition (SVD). From the result, the Left Singular Vector is obtained. Finally, to reduce data from n-dimensions to k-dimensions, first k columns are taken from the resultant vector.

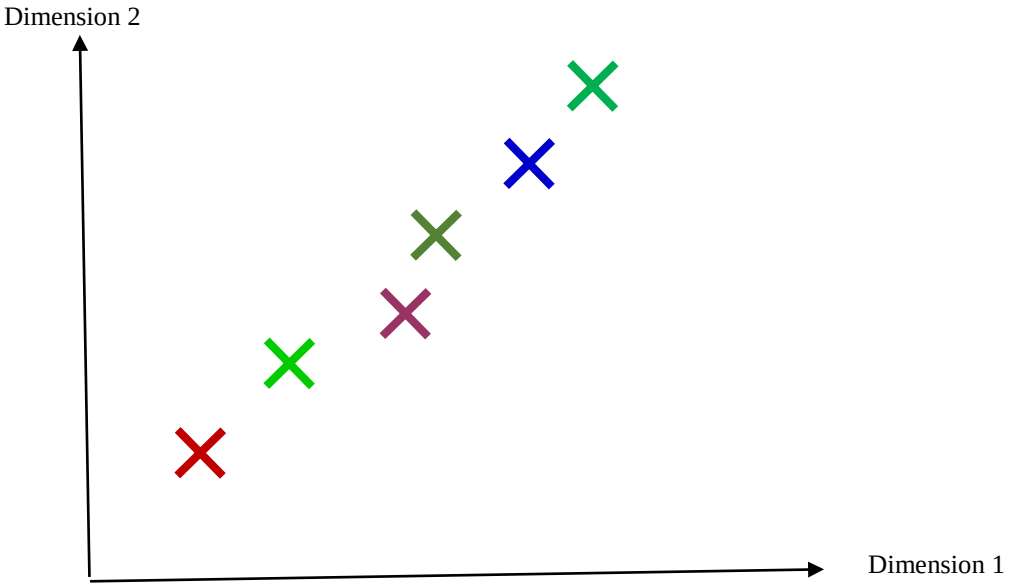

Figure 2.2 (a)

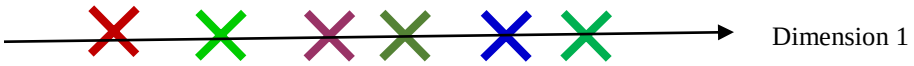

Figure 2.2 (b)

Figure 2. 2 (a) Dataset in 2D Space, (b) Dataset reduced from 2D to 1D Space

2.5.1 Mean Normalization

Before applying Principal Component Analysis, it is ideal to have performed mean normalization on the dataset beforehand. As the prime motive of PCA is to retain the maximum variance in the dataset, having mean normalization performed on the dataset leads to an efficient output; if not, it may wrongfully assume the number of components that explains all the variance in the data, thereby, perform badly.

$$\mu_j = \frac{1}{m}\sum_{i=1}^{m} x_j^{(i)} \quad (1)$$

Replace each $x_j^{(i)}$ with x_j - μ_j

- μ_j: Mean of feature
- m: Number of elements in the training set.

- $x^{(i)}$: i^{th} element in the training set

2.5.2 Covariance Matrix

A covariance matrix is a square matrix of numbers that attempts to describe the variance of the data & the covariance among variables, an empirical description of data which is being observed. Simply put, covariance answers the question: do the variables change together? For instance, among two variables under observation, if one remains stationary while the other tends to move, we can infer that the two variables are not correlated. It provides us with insights on how the variables move together by the positive, negative or non-existent character of their covariance.

Computation of Covariance Matrix:

$$\mathbf{\Sigma} = \frac{1}{m}\sum_{i=1}^{n}(x^{(i)})(x^{(i)})^{T} \quad (2)$$

2.5.3 Eigenvectors of Matrix Σ

One of the most popular used kinds of matrix decomposition is Eigen-Decomposition, where a matrix is decomposed into a set of eigenvectors & eigenvalues. Eigenvectors are axes along which the linear transformation acts. As they refine the axes of principal force that a matrix moves input along, they are useful in matrix decomposition; i.e. the diagonalization of a matrix along its eigenvectors.

An n x n matrix may have n eigenvectors, each one representing its line of action in one dimension.

Computation of Eigenvectors of matrix Σ:

$$[\mathbf{U}, \mathbf{S}, \mathbf{V}] = svd(\Sigma) \quad (3)$$

where U is a unitary matrix, S is a rectangular diagonal matrix with non-negative real numbers on the diagonal & V is a unitary matrix.

2.5.4 Algorithm

To reduce data from n-dimensions to k-dimensions

1. Computation of Covariance Matrix Σ from the input matrix.
2. Computation of Eigenvectors and Eigenvalues of Σ.

3. Selection of first k vectors from the U matrix which are the K directions onto which to project the data.

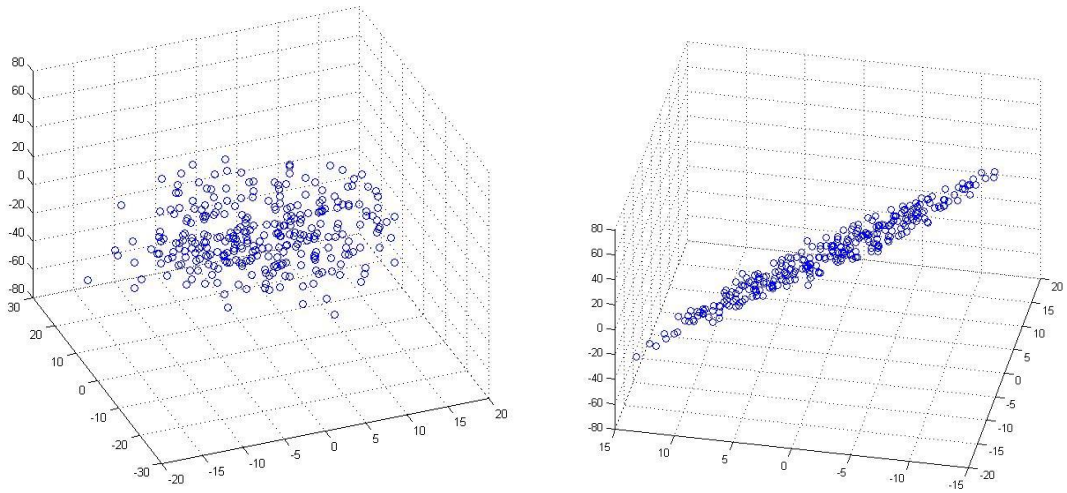

Figure 2.3 (a)

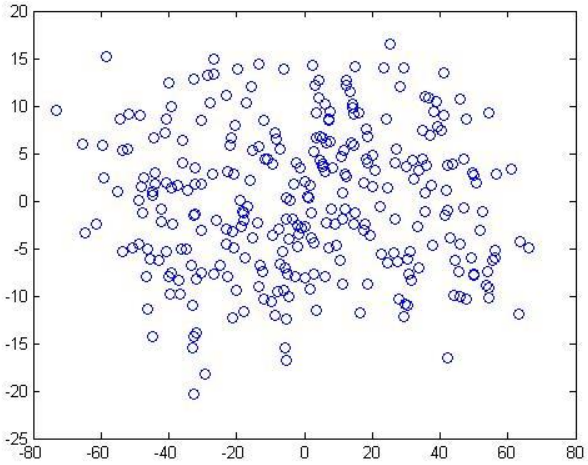

Figure 2.3 (b)

Figure 2. 3 (a) Dataset in 3D Space, (b) Dataset Reduced from 3D to 2D Space

2.6 K-Means Clustering

The K-means algorithm is classified as an unsupervised learning method of Machine Learning. The K-means clustering algorithm attempts to split a given unlabeled dataset into a fixed number of clusters. It basically incorporates two steps, namely 1) Cluster Assignment step & 2) Move centroid step.

After initializing clusters randomly, the distance from each data to the cluster centroids is measured & then the cluster is assigned to the data where the distance is minimum. Being done with that, the cluster centroid will be moved into the next location which depends on the average of the data that have been assigned to the cluster centroid previously.

These two steps are performed iteratively until K-means converges.

The inputs for the K-Means Clustering algorithm would be:

- K: Number of clusters such that $\mu_1, \mu_2, \ldots, \mu_K \in \mathbf{R}$
- Training set $\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$

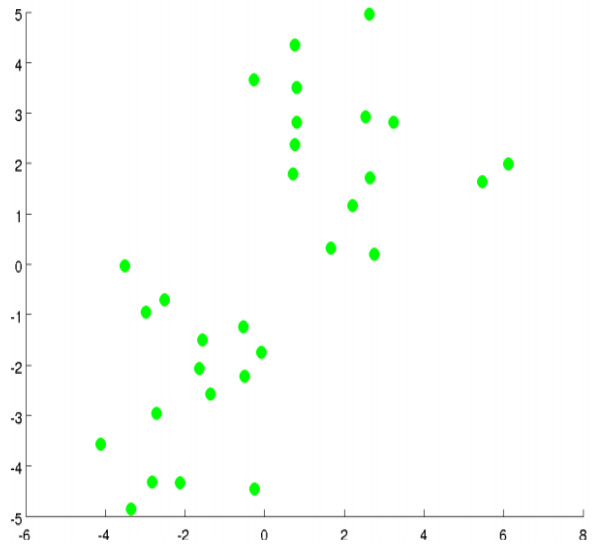

Figure 2.4 (a)

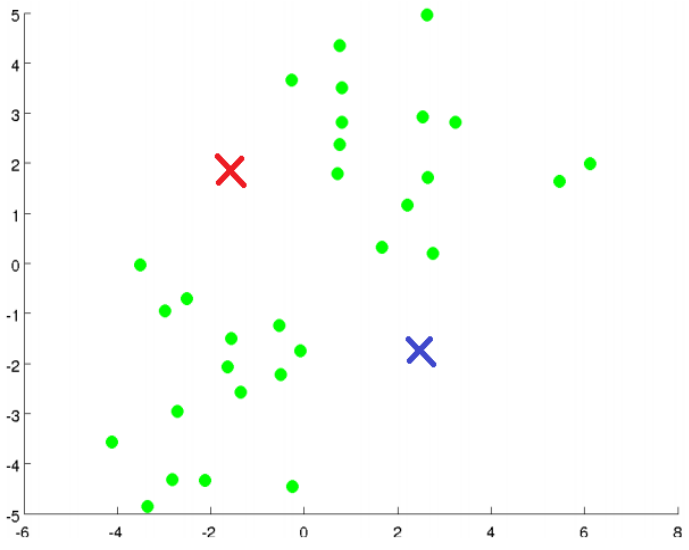

Figure 2.4 (b)

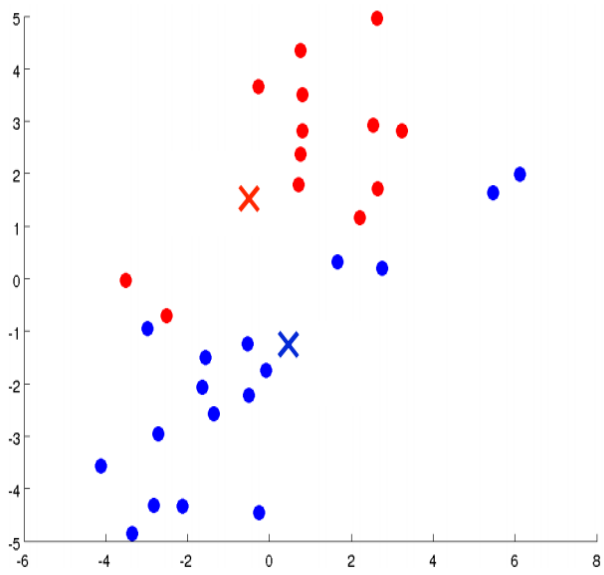

Figure 2.4 (c)

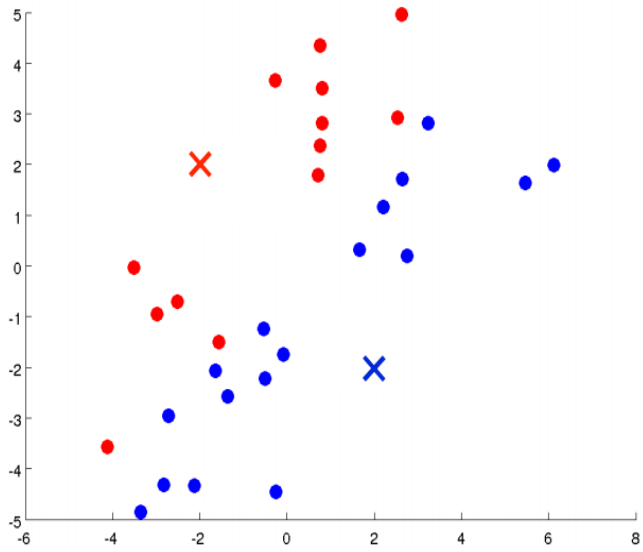

Figure 2.4 (d)

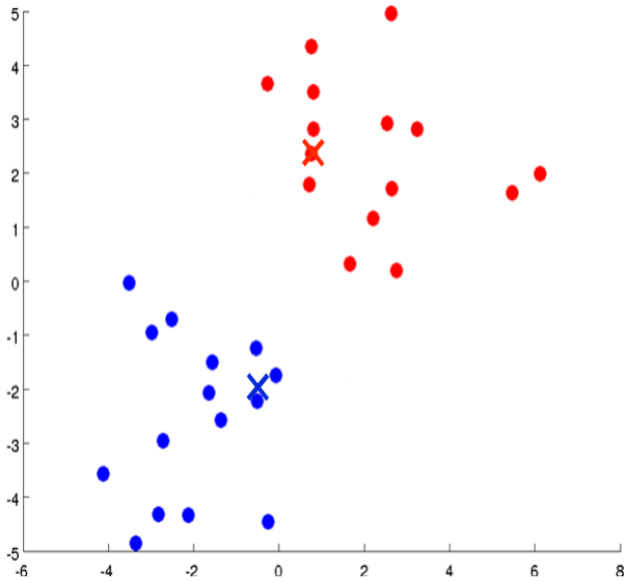

Figure 2.4 (e)

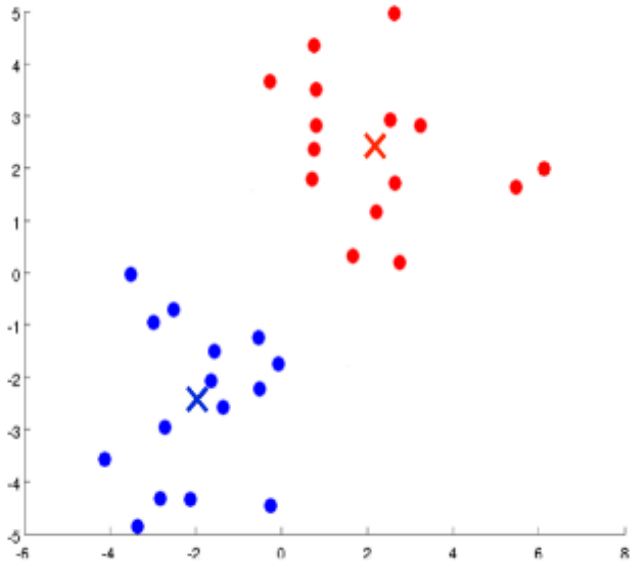

Figure 2.4 (f)

Figure 2. 4(a) Sample Dataset in 2D Space, (b) Cluster Assignment, (c) Move Centroid, (d) Cluster reassignment, (e) Move Centroid, (f) Cluster re-assignment

2.6.1 Cluster Assignment Step

Each centroid refers to one of the clusters. The initial processing involves assigning each data point in the training set to its nearest centroid, based on squared Euclidean distance. Formally put, if C_i is the collection of the centroids in set C, then each data point x is assigned to a cluster based on

$$\min_{c_i \in C} dist\,(c_i, x)^2 \tag{4}$$

where *dist* (.) is the standard Euclidean distance. Let the set of data point assignments for each i^{th} cluster centroid be S_i.

2.6.2 Move Centroid Step

After being done with the cluster assignment step, the next task in the process involves the computation of the centroids. The mean of all data points assigned to a centroid's cluster is calculated, which is the ground of this step.

$$\mu_k = average\ of\ points\ assigned\ to\ cluster\ k$$

2.6.3 Choosing number of clusters K

The algorithm is initiated with a predefined number of clusters *K.* In order to find the number of clusters, the K-Means algorithm is to be run for a range of K values & from the comparison of the results, the right number of clusters is defined. Since there is no general method to obtain the right number of clusters, but an empirical estimation can be retrieved from applying a method which is called Elbow Method. By plotting the average within-cluster distance to centroid against the different number of clusters, the point where the rate of decrease sharply shifts can be obtained and be used roughly to determine *K.*

2.6.4 Algorithm

1. Random selection of K cluster centroids.
2. Calculation of the distance between each data point & cluster centroids.
3. Assignment of the data point to the cluster centroid whose distance from the cluster centroid is minimum of all the cluster centroids.
4. Recalculation of the new cluster centroids.

5. Recalculation of the distance between each data point & new obtained cluster centroids.
6. If no data point gets reassigned, then K-means converges, else repeat from step 3.

Conclusion

In this chapter, a broad overview of the technical terms involved in the dissertation work has been represented including related explanations & equations to help better understand the proposed method.

Chapter 3

Methodology

3.1 Introduction

This chapter will illustrate a complete view of the methodology of this dissertation work. The proposed framework consists of two main stages: (1) Dimensionality Reduction using PCA (2) Color Quantization using K-means.

3.2 Proposed Methodology

3.2.1 Flow Chart

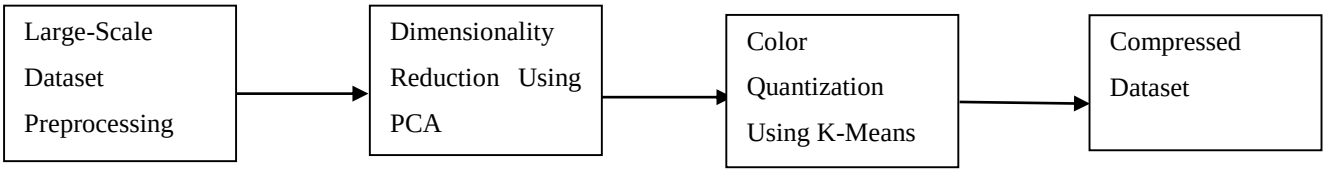

Figure 3. 1 Schematic Diagram of Proposed System

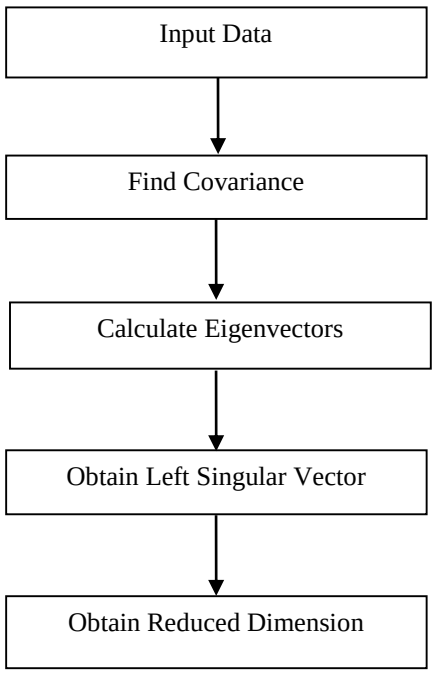

Figure 3. 2 Flow Chart of Principal Component Analysis Algorithm

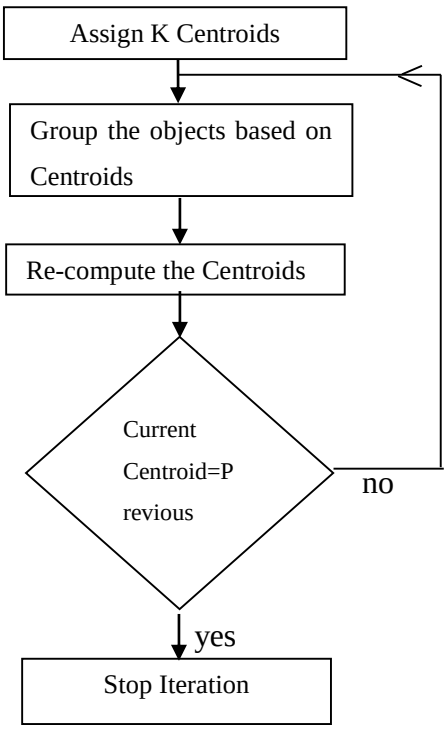

Figure 3. 3 Flow Chart of K-means Algorithm

3.2.2 Steps

The steps involved in the proposed method are explained below.

Step 1 Dimensionality Reduction using Principal Component Analysis:

- Large-scale image dataset taken as input forming an input matrix
- Computation of Covariance Matrix Σ from the input matrix.
- Computation of Eigenvectors and Eigenvalues of Σ.
- Selection of first k vectors from the Left Singular Vector in order to project the data into a lower dimensional space of k dimensions.

Step 2 Color Quantization using K-means Clustering:

- Output of the application of Principal Component Analysis taken as input forming an input matrix.
- Initialization of K cluster centroids.
- Calculation of the distance between each data point & cluster centroids.
- Assignment of the data point to the cluster centroid whose distance from the cluster centroid is minimum of all the cluster centroids.
- Recalculation of the new cluster centroids.
- Recalculation of the distance between each data point & new obtained cluster centroids.
- If no data point gets reassigned, then K-means converges.

3.3 Dimensionality Reduction

3.3.1 Overview

Dimensionality reduction is obtained by applying Principal Component Analysis on the input images. It provides the path to reduce high dimensionality, or, redundancy in other words of the images & project them onto a lower dimensional space. With different number of dimensions set for the input images, as the number of dimensions decreases, the image quality and image file size decreases along.

3.3.2 Implementation

Figure 3.4 (a)

Figure 3.4 (b)

Figure 3.4(c)

Figure 3.4 (d)

Figure 3. 4 (a) Input Image, Output of PCA (b) no. of dimensions = 35, (c) no. of dimensions = 25, (d) no. of dimensions = 15

3.4 Color Quantization

3.4.1 Overview

Color quantization is obtained by applying K-means Clustering algorithm on the input images. It provides a path to limit the number of distinct colors to a predefined value of K.

3.4.2 Implementation

Figure 3.5(a)

Figure 3.5 (b)

Figure 3.5 (c)

Figure 3.5(d)

Figure 3. 5 (a) Input Image, Output of K-means (b) no. of cluster = 256, (c) no. of cluster = 128, (d) no. of cluster = 64

Conclusion

In this chapter, each step of the proposed method is represented along with sample input & output for each step. Output for different number of dimensions & clusters have been included in order to sketch a visual difference of the two variables.

Chapter 4

Experimental Result & Analysis

4.1 Introduction

In this section, analysis of the result of the dissertation work will be represented.

4.2 Analysis of Proposed Method

Principal Component Analysis is a dimensionality reduction algorithm which substantially reduces the file size of the output images. K-means Clustering algorithm helps quantization of color, i.e. the distinct number of colors presented in the output image, which practically lessens the output file size depending on the number of clusters chosen. The general objective of this dissertation work was to obtain better compressed image output by applying Principal Component Analysis on input images, followed by color quantization by implementing K-means Clustering algorithm on the output of the Principal Component Analysis.

To obtain efficient result as proposed and better perception, different number of clusters along with different number of dimensions have been applied.

4.3 Sample Input Output

Figure 4.1 (a)

Figure 4.1 (b)

Figure 4.1 (c)

Figure 4. 1 Processing example of image dataset compression (a) Sample Input (11.11 KB) (b) Output (7.32 KB) obtained after applying PCA where number of dimension is 25 (c) Output (7.24 KB) obtained after applying K-means where number of cluster is 1024

Figure 4.2(a)

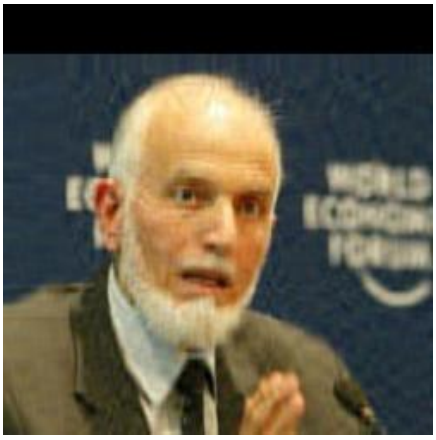

Figure 4.2 (b)

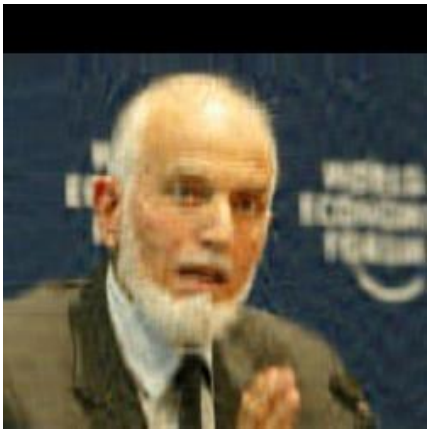

Figure 4.2 (c)

Figure 4. 2 Processing example of image dataset compression (a) Sample Input (14.4 KB) (b) Output (8.76 KB) obtained after applying PCA where number of dimension is 25 (c) Output (8.54 KB) obtained after applying K-means where number of cluster is 1024

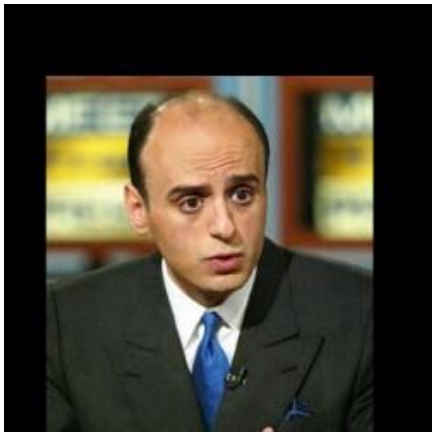

Figure 4.3(a)

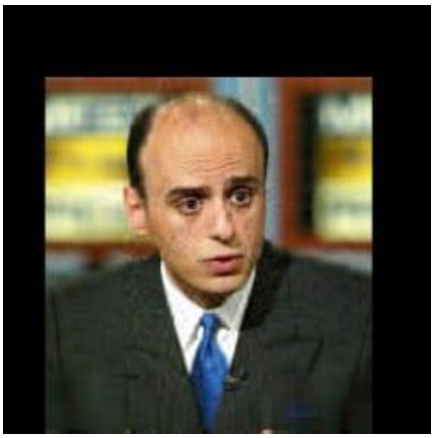

Figure 4.3 (b)

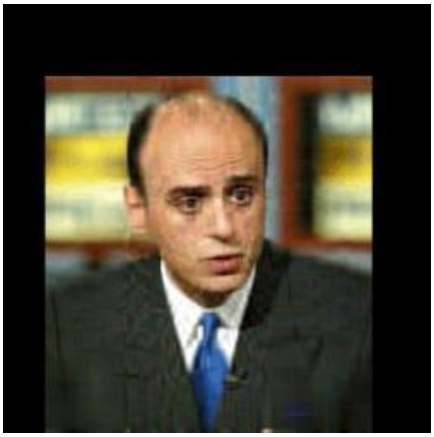

Figure 4.3 (c)

Figure 4. 3 Processing example of image dataset compression (a) Sample Input (12.2 KB) (b) Output (7.52 KB) obtained after applying PCA where number of dimension is 25 (c) Output (7.41 KB) obtained after applying K-means where number of cluster is 1024

Figure 4.4 (a)

Figure 4.4 (b)

Figure 4.4 (c)

Figure 4. 4 Processing example of image dataset compression (a) Sample Input (14.4 KB) (b) Output (8.45 KB) obtained after applying PCA where number of dimension is 25 (c) Output (8.24 KB) obtained after applying K-means where number of cluster is 1024

4.4 Procedure & Evaluation Parameters

The system performance has been evaluated on the basis of Compression Ratio, Mean Squared Error (MSE) & Peak Signal to Noise Ratio (PSNR).

Compression Ratio: It is defined as the ratio between the uncompressed file size & compressed file size.

$$Compression\ Ratio = \frac{Uncompressed\ File\ size}{Compressed\ File\ size}$$

Mean Squared Error (MSE): The mean squared error is the cumulative squared error between the compressed & the original image. The mathematical formulae for MSE is:

$$MSE = \frac{1}{MN}\sum_{y=1}^{M}\sum_{x=1}^{N}[\ I(x,y) - I'(x,y)]^2 \qquad (5)$$

where $I(x,y)$ is the original image, $I'(x,y)$ is the compressed image & $M,\ N$ are the dimensions of the images. A lower value for MSE means lesser error & a higher value implies greater error.

Peak Signal to Noise Ratio (PSNR): It is a measure of the peak error. The mathematical formulae for PSNR is:

$$PSNR = 10\log_{10}(\frac{255^2}{MSE}) \qquad (6)$$

As it has an inverse relationship with MSE, a higher value of PSNR is better as it means that the ratio of Signal to Noise is higher, where the Signal refers to the original image & the noise refers to the error in reconstruction.

4.5 Experimental Result Analysis:

4.5.1 Proposed Method vs K-means

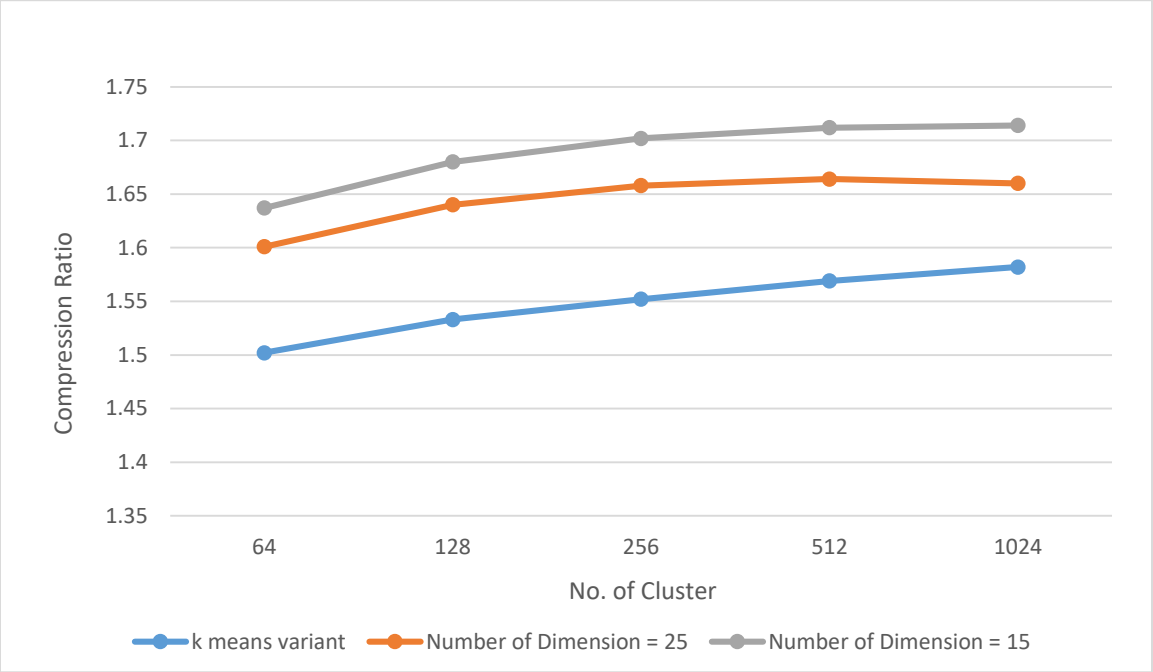

Figure 4. 5 Number of Clusters versus Compression Ratio

In this graphical representation, the X axis is assigned with different number of clusters & Y axis is assigned with Compression Ratio. Compression ratio for different number of clusters obtained from K-means is represented with the Blue line. The proposed method, where number of dimension is taken 25 & 15 is represented with the Orange line & Silver line respectively.

For number of cluster = 256, the compression ratio from K-means is 1.552. In the proposed method, for number of cluster = 256, when no. of dimension = 25, the compression ratio increases to 1.658 & when number of dimension = 15, the compression ratio increases to 1.702. The compression ratio increases as the number of clusters for color quantization increases.

4.5.2 Proposed Method vs PCA

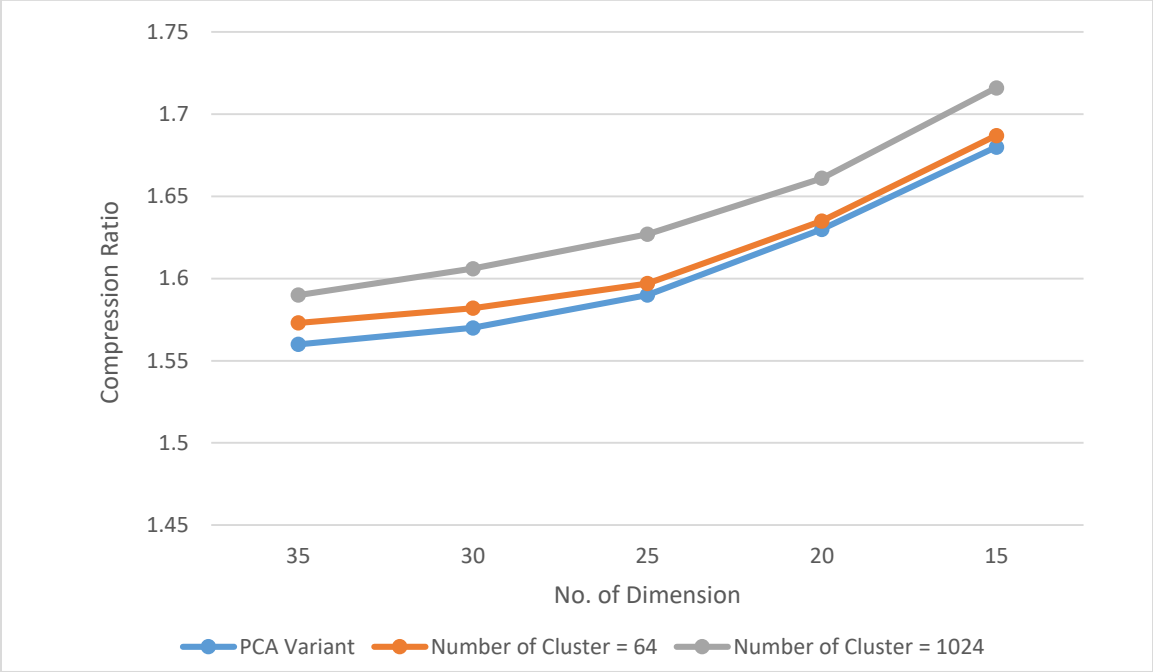

Figure 4. 6 Number of Dimension versus Compression Ratio

In this graphical representation, the X axis is assigned with different number of dimensions & Y axis is assigned with Compression Ratio. Compression ratio for different number of dimensions obtained from PCA is represented with the Blue line. The proposed method, where number of cluster is taken 64 & 1024 is represented with the Orange line & Silver line respectively.

For number of dimension = 25, the compression ratio from PCA is 1.59. In the proposed method, for number of dimension = 25, when number of cluster = 64, the compression ratio increases to 1.59 & when number of cluster = 1064, the compression ratio increases to 1.63. The compression ratio increases as the number of dimension decreases, as the number of dimension represents the lower dimensional space onto which we project our data.

4.5.3 K-means vs PCA vs Proposed Method

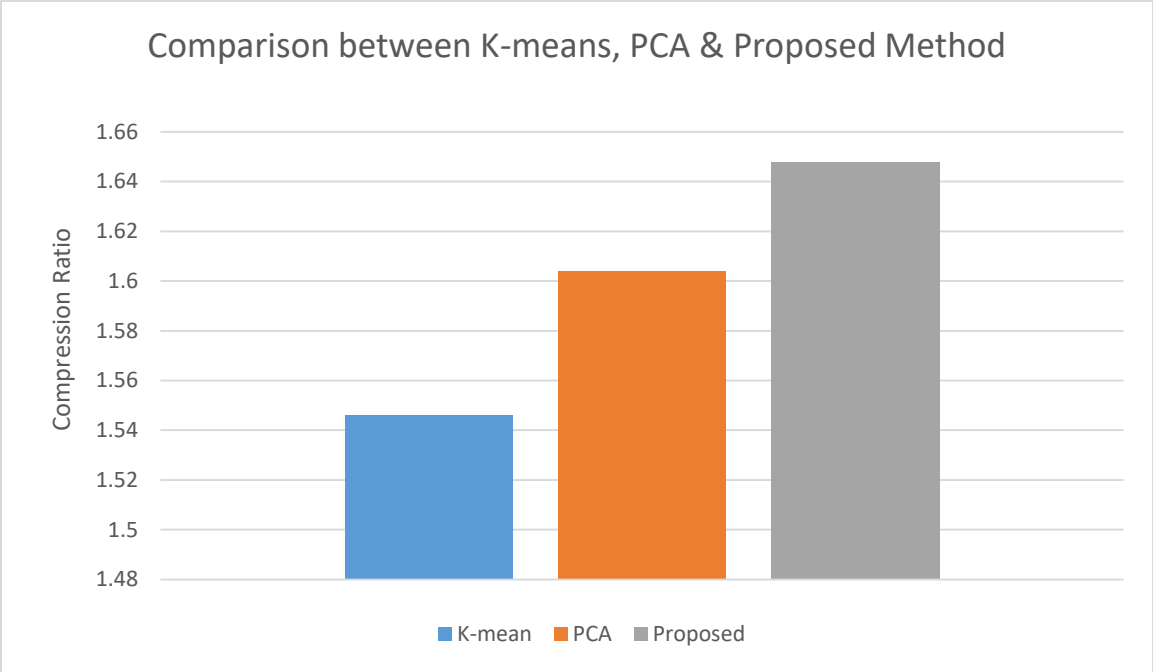

Figure 4. 7 Comparison between K-means, PCA & Proposed Method

In this graphical representation, the K-means, PCA & Proposed Method have been plotted where the average compression ratio for the large-scale dataset is obtained 1.543 from K-means, 1.612 from PCA & 1.644 from the proposed method.

4.5.4 Mean Squared Error

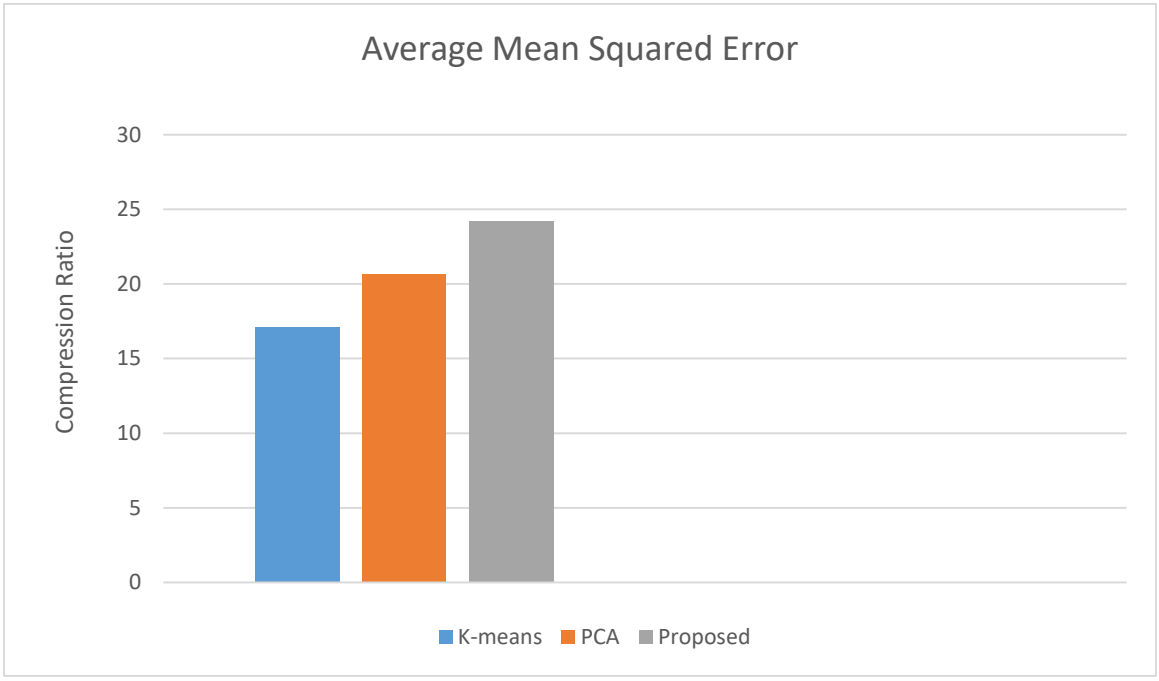

Figure 4. 8 Comparison of MSE of K-means, PCA & Proposed Method

As more compression ratio is being obtain, the compressed image permanently loses some of its data, as for which, the MSE value increases with the increase in compression ratio denoting the loss in image data.

4.5.5 Peak Signal to Noise Ratio

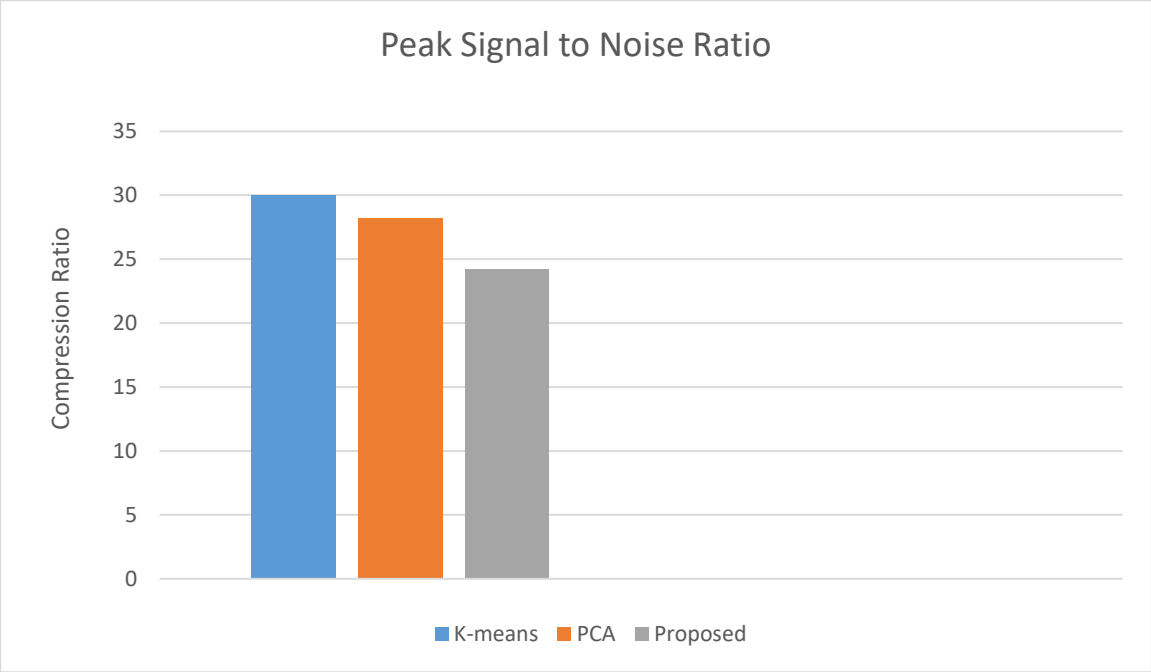

Figure 4. 9 Comparison of PSNR of K-means, PCA & Proposed Method

Since, the more an image is compressed, the more it permanently loses data, with the increase in compression ratio, the peak signal to noise ratio tends to decrease.

Conclusion

In this chapter, an in-depth comparison of the K-means, PCA & proposed method have been performed & represented graphically in order to obtain a perception that in similar cases, the proposed method performs more efficiently than implementing K-means & PCA alone.

Chapter 5

Conclusion & Future Works

5.1 Conclusion

In this dissertation work, a new method for compression of a large-scale image dataset is proposed with the goal to better compress a large-scale image dataset. Initially, dimensionality reduction is performed by implementing Principal Component Analysis. Following that, color quantization is performed by implementing K-means clustering algorithm.

The main contribution of this dissertation work is to obtain a better compression ratio for a large-scale image dataset in order to reduce data redundancy by performing dimensionality reduction & color quantization. Compression ratio obtained from implementing PCA & K-means alone is less than that obtained from implementing the proposed method. While performing the proposed method, the two factors that are to be kept in consideration is the no. of dimensions while performing PCA & the no. of clusters while performing K-means clustering algorithm. As the no. of dimension reduces, the compression ratio increases as we project the data onto a lower dimensional space. As the no. of cluster increases, the compression ratio increases due to better color quantization performance.

5.2 Future Works

Before finalizing the thesis work, we mention here with some brief remarks on future extensions of our work presented.

The cluster initialization in the proposed method is done by random initialization which sometimes leads to converge at a local optima. To overcome this issue, K-means++ algorithm can be implemented which solves the issue of the algorithm converging at a local optima & help find the global optima.

The proposed framework can be a preprocessing step to perform human facial recognition with a human face dataset, animal detection by training the model with an animal dataset, object detection & recognition & action recognition.

Bibliography

[1] Žliobaitė, Indrė, et al. "Active learning with evolving streaming data." Machine Learning and Knowledge Discovery, in *Databases. Springer Berlin Heidelberg,* 2011. 597–612.

[2] Genevieve C. Ngo & Erees Queen B. Macabebe, "Image Segmentation Using K-Means Color Quantization & Density-Based Spatial Clustering of Applications with Noise (DBSCAN) for Hotspot Detection in Photovoltaic Modules", in *IEEE Region 10 Conference (TENCON) — Proceedings of the International Conference,* 2016.

[3] Herve Abdi & Lynne J. Williams, "Principal Component Analysis", in *2010 John Wiley & Sons, Inc. WIREs Comp Stat* 2010 2 433–459.

[4] Nitika Sharma, & Kriti Saroha, "A Novel Dimensionality Reduction Method for Cancer

Dataset using PCA and Feature Ranking", in *International Conference on Advances in Computing, Communications and Informatics (ICACCI)* 2015.

[5] Chih-Wen & Jyh-Horng, "Image compression using PCA with clustering", in *IEEE International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS 2012)* November 4-7, 2012

[6] Turgay Celik, "Unsupervised Change Detection in Satellite Images Using Principal Component Analysis and K-Means Clustering", in *IEEE GEOSCIENCE AND REMOTE SENSING LETTERS,* VOL. 6, NO. 4, OCTOBER 2009.

[7] R. C. Gonzalez and R. E. Woods, "Digital Image Processing," *3rd ed. Upper Saddle River, NJ: Prentice-Hall,* 2006.

Appendix A

Source Code

```
%% K-means Clustering
clear all;
close all;
clc;

facematrixred = zeros(100, 250 * 250);
facematrixgreen = zeros(100, 250 * 250);
facematrixblue = zeros(100, 250 * 250);

fprintf('\nRunning K-Means clustering on pixels from an image.\n\n');
imagefiles = dir('LenaPCA25/*.jpg');
nfiles = length(imagefiles);
sprintf('nfiles %d .', nfiles)
for ii=1:nfiles
    currentfilename = imagefiles(ii).name;
    A = double(imread(strcat('images/',currentfilename)));
    A = A / 255;
    img_size = size(A);


    X = reshape(A, img_size(1) * img_size(2), 3);
    K = 64;
    max_iters = 100;
    initial_centroids = kMeansInitCentroids(X, K);

    [centroids, idx] = runkMeans(X, initial_centroids, max_iters);


    idx = findClosestCentroids(X, centroids);
    X_recovered = centroids(idx,:);

    X_recovered = reshape(X_recovered, img_size(1), img_size(2), 3);

    subplot(1, 2, 1);
    imagesc(A);
    title('Original');

    subplot(1, 2, 2);
    imagesc(X_recovered)
    title(sprintf('Compressed, with %d colors.', K));

    folder = 'D:\Notes\Thesis\Rayan\pcakmeans1024';
    fullFileName = fullfile(folder, strcat('', currentfilename));
    imwrite(X_recovered, fullFileName);
    X = reshape(X_recovered, img_size(1) * img_size(2), 3);
    facematrixred(ii, :) = X(:,1)';
    facematrixgreen(ii, :) = X(:,2)';
    facematrixblue(ii, :) = X(:,3)';
```

```
end

save('faces3.mat','facematrixred','facematrixgreen','facematrixblue');

function centroids = kMeansInitCentroids(X, K)


centroids = zeros(K, size(X, 2));

randidx = randperm(size(X, 1));

centroids = X(randidx(1:K), :);

end

function idx = findClosestCentroids(X, centroids)
k = size(centroids, 1);
idx = zeros(size(X,1), 1);

dist = zeros(size(X,1), k);
for cntr = 1 : k
    temp = bsxfun(@minus, X , centroids(cntr, :));
    dist(:,cntr) = sum(temp.^2, 2);
end

[Y,idx] = min(dist, [], 2);

end


%%Principal Component Analysis

imagefiles = dir('images/*.jpg');

for j=1:100
    currentfilename = imagefiles(j).name;
    img = imread(strcat('images/',currentfilename));
    %%grayscale image
    figure(1);

    subplot(1,2,1);
    imshow(img);
    title('Original');

    %%SVD on grayscale NxN matrix
    shape = size(img);
    U = zeros(shape);
    S = zeros(shape);
    V = zeros(shape);
    for i = 1:3
        [U(:,:,i),S(:,:,i),V(:,:,i)] = svd(im2double(img(:,:,i)));
    end
```

```
    %% Contains sub-portion of components
    Ug = zeros(shape);
    Sg = zeros(shape);
    Vgt = zeros(shape);
    Vt = zeros(shape);
    for i = 1:3
        Vt(:,:,i) = V(:,:,i)';
    end

    %% Regenerate from components (for varying sub-portions p)
    % N = size(img,1);
    N = 100;
    p = 25;

    regenerated_img = zeros(shape);

    for i = 1:3
        Ug(:, 1:p, i) = U(:, 1:p, i);
        Sg(1:p, 1:p, i) = S(1:p, 1:p, i);
        Vgt(1:p, :, i) = Vt(1:p, :, i);
        regenerated_img(:,:,i) = Ug(:,:,i) * Sg(:,:,i) * Vgt(:,:,i);
    end

    folder = 'D:\Notes\Thesis\Rayan\LenaPCA25';
    fullFileName = fullfile(folder, strcat('', currentfilename));
    imwrite(regenerated_img, fullFileName);
end

%%Mean Squared Error & Peak Signal to Noise Ratio
clc;
close all;
clear;
workspace;
format long g;
format compact;
fontSize = 20;



grayImage = imread('images/10.jpg');
[rows columns] = size(grayImage);


subplot(2, 2, 1);
imshow(grayImage, []);
title('Original Gray Scale Image', 'FontSize', fontSize);
set(gcf, 'Position', get(0,'Screensize'));


noisyImage = imread('pcaonlyp25/pcacopy10.jpg');

subplot(2, 2, 2);
imshow(noisyImage, []);
title('Noisy Image', 'FontSize', fontSize);
```

```
squaredErrorImage = (double(grayImage) - double(noisyImage)) .^ 2;

subplot(2, 2, 3);
imshow(squaredErrorImage, []);
title('Squared Error Image', 'FontSize', fontSize);


mse = sum(sum(squaredErrorImage)) / (rows * columns);

PSNR = 10 * log10( 256^2 / mse);

message = sprintf('The mean square error is %.2f.\nThe PSNR = %.2f\n', mse,
PSNR);
msgbox(message);
```