**Image Compression using PCA and K-Means Clustering**

THE PROJECT REPORT

***Submitted by***

BL.EN. U4AIE19028 K.Vishnu Sainadh

BL.EN. U4AIE19034 K.Satwik

BL.EN. U4AIE19066 V.Ashrith

**For the course**

# 19MAT212- Mathematics and Intelligent Systems - 4

**Guided and Evaluated by**

***Dr. Subramani R & Dr.Murali K***

***Assistant Professor***



AMRITA SCHOOL OF ENGINEERING, BANGALORE

AMRITA VISHWA VIDYAPEETHAM

**BANGALORE 560 035**

May-2021

**AMRITA VISHWA VIDYAPEETHAM**

Title of the project

**Image Compression using PCA and K-Means Clustering**

**Acknowledgement**

The satisfaction that accompanies successful completion of any task would be incomplete without mention of people whom made it possible, and whose constant encouragement and guidance have been source of inspiration throughout the course of this project work.

We offer our sincere pranamas at the lotus feet of “AMMA”, MATA AMRITANANDAMAYI DEVI who showered her blessing upon us throughout the course of this project work.

It is a great pleasure to express our gratitude and indebtedness to our project guide Dr.Subramani R & Dr.Murali K , Assistant Professor, Amrita School of Engineering, Bangalore for your valuable guidance, encouragement, moral support and affection throughout the semester.

#### TABLE OF CONTENTS:

1). ABSTRACT……………...………………………….4

2). INTRODUCTION ….………………………………4

3). CONCEPTS USED………………………………….4

* Principal Component Analysis (PCA) ……….4
* K-Means Clustering ………………………….6

4). RESULTS ……………………………………………8

5). CONCLUSION …………………………………….10

6). FUTURE SCOPE …………………………………..10

7). REFERENCES …………………………………….10

**Abstract :**

The image has touched several areas of our lives, so we need to have well-shaped images with less and less sizes. We are comparing image compression techniques based on Principal Component Analysis (PCA) and K-means Clustering. PCA is based on the reduction of the image vectors of the image using the principal ones with multiple factors. K-Means clustering algorithm reduces the number of colors so that it only needs to store certain numbers of RGB values. Thus, it will reduce the image size making it more efficient in the storage.

**Introduction :**

The image is a very powerful information medium; it touched several areas of our life: medicine, weather, telecommunications, cartography, geology, etc. With the development of the computer tool, several image processing techniques have emerged. Among the many treatments that can be performed on images, there is the image compression operation.

Image compression is the process of encoding digital image information using fewer bits than an unencoded representation would use through use of specific encoding schemes. In our project we are going to implement and compare the result of K-Means clustering algorithm and Principal Component Analysis (PCA) on image compression using the sklearn package. Compressed images are evaluated based on the memory size reduction and percentage on how good it can summarize or explain the variance of the original image. The objective of image compression is to reduce the memory size to be as small as possible while maintaining the similarity with the original image, which is represented by the explained variance percentage.

**Concepts used :**

**Principal Component Analysis (PCA) :**

It is a way of identifying patterns in data, and expressing the data in such a way as to highlight their similarities and differences. Since patterns in data can be hard to find in data of high dimension, where the luxury of graphical representation is not available, PCA is a powerful tool for analysing data.

The other main advantage of PCA is that once you have found these patterns in the data, you compress the data, ie. by reducing the number of dimensions, without much loss of information.

**Method :**

STEP 1: Subtract the mean

For PCA to work properly, you have to subtract the mean from each of the data dimensions. The mean subtracted is the average across each dimension. So, all the X values have X(bar) (the mean of the X values of all the data points) subtracted, and all the Y values have Y(bar) subtracted from them. This produces a data set whose mean is zero.

STEP 2: Calculate the covariance matrix

A covariance matrix is a square matrix giving the covariance between each pair of elements of a given random vector.

STEP 3: Calculate the eigenvectors and eigenvalues of the covariance matrix

Since the covariance matrix is square, we can calculate the eigenvectors and eigenvalues for this matrix. These are important, as they tell us useful information about our data. It is important to notice that these eigenvectors are both unit eigenvectors ie. their lengths are both 1.

STEP 4: Choosing components and forming a feature vector

Here is where the notion of data compression and reduced dimensionality comes into it. The eigenvector with the highest eigenvalue is the principle component of the data set. The eigenvector with the largest eigenvalue was the one that pointed down the middle of the data. It is the most significant relationship between the data dimensions.

In general, once eigenvectors are found from the covariance matrix, the next step is to order them by eigenvalue, highest to lowest. This gives you the components in order of significance. Now, if you like, you can decide to ignore the components of lesser significance. You do lose some information, but if the eigenvalues are small, you don’t lose much. If you leave out some components, the final data set will have less dimensions than the original. To be precise, if you originally have n dimensions in your data, and so you calculate n eigenvectors and eigenvalues, and then you choose only the first p eigenvectors, then the final data set has only p dimensions.

Now we need to form a feature vector. This is constructed by taking the eigenvectors that you want to keep from the list of eigenvectors, and forming a matrix with these eigenvectors in the columns.

STEP 5: Deriving the new data set

This is the final step in PCA. Once we have chosen the components (eigenvectors) that we wish to keep in our data and formed a feature vector, we simply take the transpose of the vector and multiply it on the left of the original data set, transposed.



Where RowFeatureVector is the matrix with the eigenvectors in the columns transposed so that the eigenvectors are now in the rows, with the most significant eigenvector at the top, and RowDataAdjust is the mean-adjusted data transposed, ie. The data items are in each column, with each row holding a separate dimension.

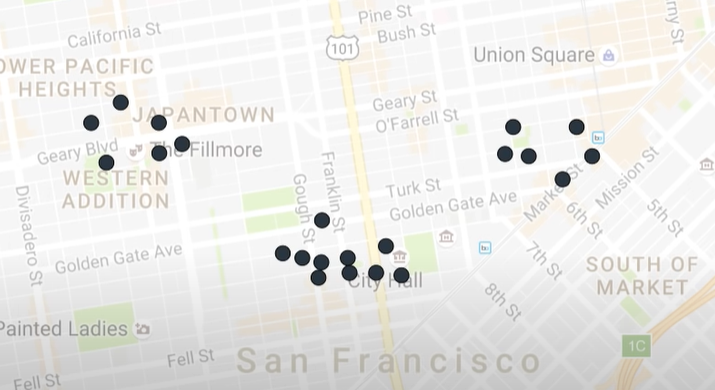
Step 6 : At last we add back the mean since we have subtracted it at the start.

**K-Means Clustering:**

K-Means Clustering is an unsupervised machine learning algorithm. In contrast to traditional supervised machine learning algorithms, K-Means attempts to classify data without having first been trained with labeled data. Once the algorithm has been run and the groups are defined, any new data can be easily assigned to the most relevant group. Using K-means clustering, we will perform quantization of colors present in the image which will further help in compressing the image.

**Method :**

Let us consider a scenario, this is a random data

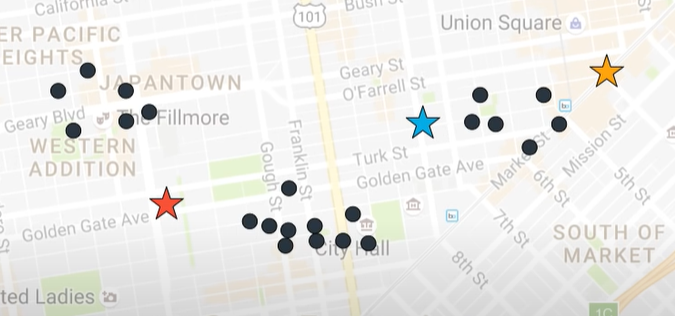


STEP 1: choose the number of k clusters

We chose 3 clusters

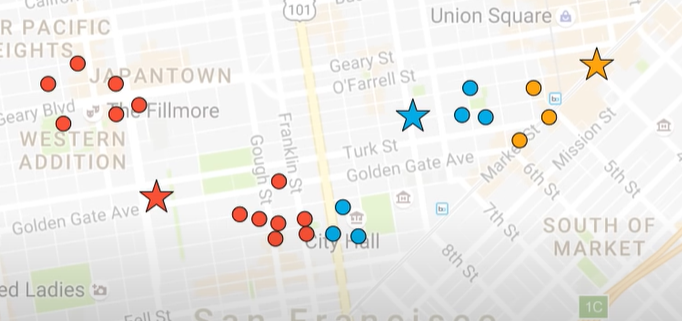
STEP 2: select at random K points

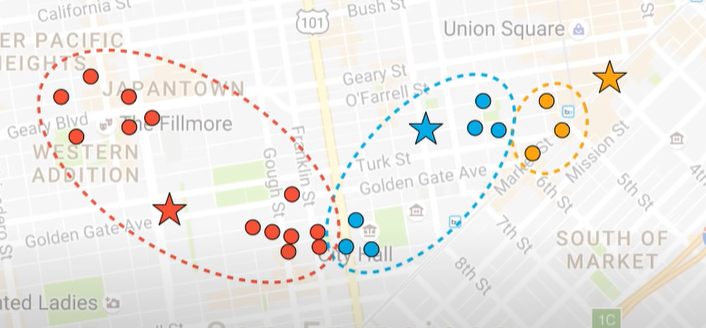
We now located 3 random points



STEP 3: Assign each data point to the closest centroid that forms k clusters

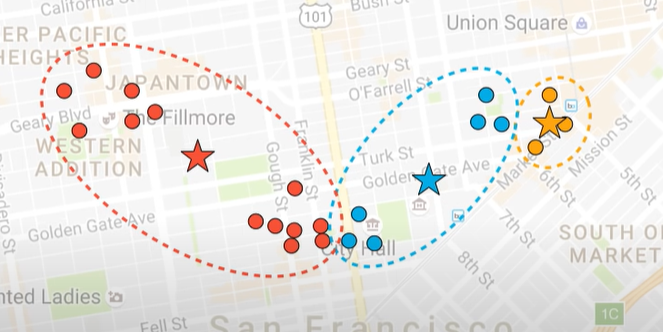
We now assigned data points to their closest centroids





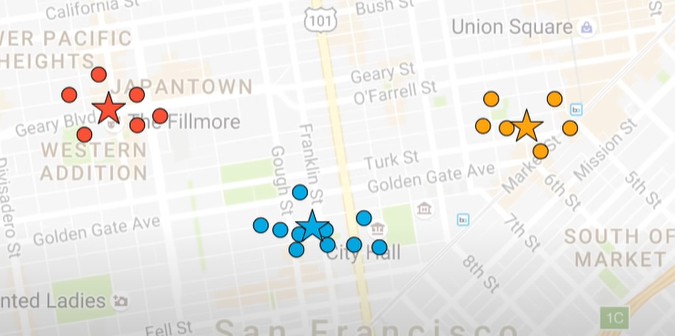
STEP 4: compute and place the new centroid of each cluster

We now changed the position of the centroid



STEP 5: reassign each data point to the new closest centroid, if any re-assignment took place go to step 4

So after reassigning data points and after changing the centroid we get

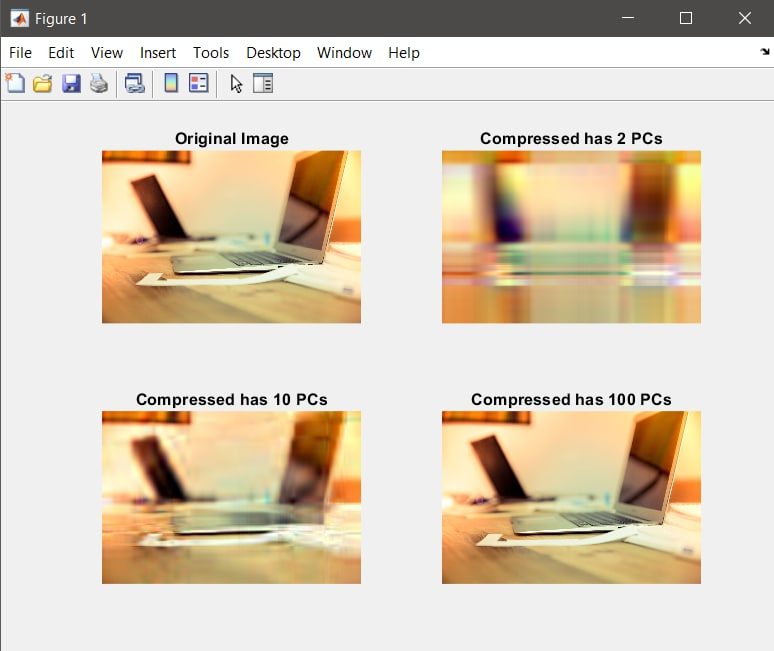


## **How does the K-Means Clustering technique compress the image?**

Each pixel in a colourful image is 3 bytes (RGB), and each colour can have intensity values ranging from 0 to 255. The total number of colours that can be represented is 256\*256\*256, according to combinatorics ( equal to 16,777,216). In fact, we can only see a few colours in an image, which is far fewer than the stated number. As a result, the k-Means Clustering method takes use of the human eye's visual perception and represents the image with a limited number of colours. Colors with varied intensity values that are RGB values appear to the human eye to be the same. The K-Means algorithm takes use of this and groups colours that appear to be similar (which are close together in a cluster).

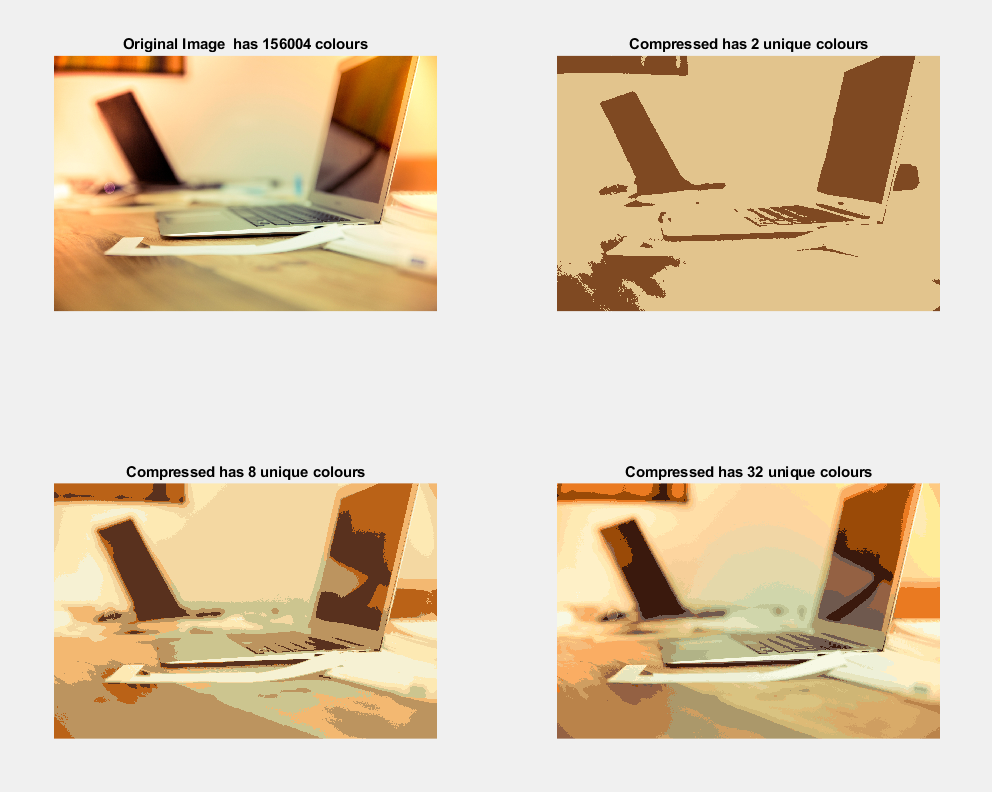
**Results:**

Image Compression using PCA



The obtained results using PCA clearly show that this technique gives good results with uncorrelated and redundant data. The compression ratio was entirely affected by the number of bits reserved to represent the real factors associated with PCA vectors.

Image Compression using K-Means Clustering



**Conclusion :**

We successfully perform image compression by using the Unsupervised Learning algorithm, such as K-Means Clustering and Dimensionality Reduction using Principal Component Analysis (PCA).

K-Means is more recommended to be used in reducing image size compared to PCA but if we want to keep the overall color of the original image, use PCA.

**Future Scope :**

What can we do to improve the image compression? One way is to perform and evaluate the two methods successively, i.e. K-Means then PCA or PCA then K-Means. From the best method, we can create an automated script to compress a massive amount of images in a folder.

**References :**

1).https://www.researchgate.net/profile/Mohammad-Mofarreh-Bonab/publication/266890475\_A\_New\_Technique\_for\_Image\_Compression\_Using\_PCA/links/57778ac908aead7ba07440dc/A-New-Technique-for-Image-Compression-Using-PCA.pdf

2). https://ieeexplore.ieee.org/abstract/document/6473533